



**CENTRO UNIVERSITÁRIO DE BRASÍLIA -UniCEUB**  
**CURSO DE ENGENHARIA DE COMPUTAÇÃO**

**CRISTIANO ARAÚJO TAVARES**

**CONTROLE DE RESFRIAMENTO DE SERVIDORES UTILIZANDO**  
**UM MICROCONTROLADOR**

**Orientadora: Prof<sup>ª</sup> MsC. Maria Marony Sousa Farias**

Brasília  
novembro, 2013

**CRISTIANO ARAÚJO TAVARES**

**CONTROLE DE RESFRIAMENTO DE SERVIDORES  
UTILIZANDO UM MICROCONTROLADOR**

Trabalho apresentado ao Centro  
Universitário de Brasília  
(UniCEUB) como pré-requisito  
para a obtenção de Certificado de  
Conclusão de Curso de Engenharia  
de Computação.

Orientadora: Prof<sup>ª</sup>. Maria Marony  
Sousa Farias

Brasília  
novembro, 2013

**CRISTIANO ARAÚJO TAVARES**

**CONTROLE DE RESFRIAMENTO DE SERVIDORES  
UTILIZANDO UM MICROCONTROLADOR**

Trabalho apresentado ao Centro  
Universitário de Brasília  
(UniCEUB) como pré-requisito  
para a obtenção de Certificado de  
Conclusão de Curso de Engenharia  
de Computação.  
Orientadora: Prof<sup>ª</sup>. Maria Marony  
Sousa Farias

Este Trabalho foi julgado adequado para a obtenção do Título de Engenheiro de Computação,  
e aprovado em sua forma final pela Faculdade de Tecnologia e Ciências Sociais Aplicadas -  
FATECS.

---

Prof. Abiezer Amarilia Fernandes  
Coordenador do Curso

**Banca Examinadora:**

---

Prof<sup>ª</sup> Maria Marony Sousa Farias, MsC.  
Orientadora

---

Prof. Fernando Chagas Santos, MsC.  
UniCEUB

---

Prof. Sidney Cerqueira Bispo dos Santos, Dr.  
UniCEUB

## **DEDICATÓRIA**

Aos meus pais, por todo o apoio e  
principalmente por acreditarem em mim.  
À minha esposa Larissa e minha filha Luíza.



## **AGRADECIMENTOS**

Agradeço a Deus por me iluminar e guiar meus passos.

Agradeço a minha amada e doce filha Luíza Ribeiro Tavares por transformar a minha vida.

Agradeço a minha esposa Larissa Vanderlei Ribeiro pela paciência, força, companheirismo e amor nos momentos mais difíceis da minha vida.

Agradeço aos meus pais, Cristovam Bezerra Tavares e Audilâne M<sup>a</sup> Araújo Tavares pelas palavras sábias, incentivo e amor incondicional.

Aos meus colegas do curso de Engenharia de Computação pela longa trajetória que percorremos juntos, ajudando uns aos outros.

Agradeço também ao corpo docente do curso de Engenharia de Computação pelo comprometimento, dedicação, e preocupação em lapidar alunos e transformá-los em engenheiros vencedores.

*"O maior inimigo do conhecimento não é a  
ignorância, mas a ilusão do conhecimento."*

(Stephen Hawking)

## SUMÁRIO

<b>CAPÍTULO 1 - INTRODUÇÃO</b>	13
<b>1.1 – Apresentação do Problema</b>	13
<b>1.2 – Objetivos do Trabalho</b>	13
1.2.1– Objetivo Geral	13
1.2.2 – Objetivos Específicos	13
<b>1.3 – Justificativa e Importância do Trabalho</b>	13
<b>1.4 – Escopo do Trabalho</b>	13
<b>1.5 – Resultados Esperados</b>	14
<b>1.6 – Estrutura do Trabalho</b>	14
<b>CAPÍTULO 2 – APRESENTAÇÃO DO PROBLEMA</b>	15
<b>CAPÍTULO 3 – BASES METODOLÓGICAS PARA RESOLUÇÃO DO PROBLEMA</b>	22
<b>3.1 – Visões da Infraestrutura de TI</b>	22
3.1.2 – Visão Geral do Datacenter	22
3.1.2.1 – Conceito	23
3.1.2.2 – Histórico	24
3.1.2.3 – Classificação de DATACENTERS em TIERS 1, 2, 3 e 4	25
3.1.2.4 –DATACENTER Verde (Green DATACENTER)	26
3.1.2.5 – Tendências	
<b>3.2 – DATACENTER Verde (Green DATACENTER)</b>	26
<b>3.3 – Hardware e Interfaces para Comunicação com o Circuito</b>	27
3.3.1 – Computador Pessoal	27
3.3.2 – Porta USB	29
<b>3.5 – O Arduino</b>	31
3.5.1 – Arquitetura	33
3.5.2 – <i>Shield Ethernet</i>	34
<b>3.6 – Componentes eletrônicos e sensores utilizados no projeto</b>	34
3.6.1 – Sensor de Temperatura e umidade DTH11	34
3.6.2 – Transistor TIP120	35
3.6.3 – Sensor de Luminosidade LDR	36
3.6.4 – Resistor	37
3.6.4 – Fans e Coolers	38
<b>3.7 – Linguagens de Programação Utilizadas no Projeto</b>	39
3.7.1 – Linguagem C	39
<b>3.8 – Meio de Transmissão das Informações</b>	40
3.8.1 – Protocolo TCP/IP	40
<b>CAPÍTULO 4 – MODELO PROPOSTO</b>	42
<b>4.1 – Apresentações Gerais do Modelo Proposto</b>	42
<b>4.2 – Descrições das Etapas do Modelo</b>	42
<b>4.3 - Descrição da Implementação</b>	45
<b>CAPÍTULO 5 - APLICAÇÃO PRÁTICA DO MODELO PROPOSTO</b>	47
<b>5.1 - Apresentação da área de Aplicação do modelo</b>	47
<b>5.2 – Descrição da Aplicação do Modelo</b>	47
<b>5.3 – Resultados da Aplicação do Modelo</b>	48
<b>5.4 – Custos do modelo proposto</b>	49
<b>5.3 – Avaliação Global do Modelo</b>	49
<b>CAPÍTULO 6 - CONCLUSÃO</b>	50
<b>6.1 – Conclusões</b>	50

<b>6.2 – Sugestões para Trabalhos Futuros.....</b>	<b>50</b>
<b>REFERÊNCIAS .....</b>	<b>51</b>
<b>APÊNDICE .....</b>	<b>53</b>

## LISTA DE FIGURAS

Figura 2.1 - Consumo Típico de Energia em um Datacenter. ....	15
Figura 2.2 - Temperatura dentro de um dos 45 Datacenters analisados. ....	17
Figura 2.3 - Temperatura dentro de um Datacenter. ....	18
Figura 2.4 - Evolução de carga térmica. ....	19
Figura 2.5 - Um modelo CFD em um Datacenter. ....	20
Figura 2.6 - <i>Hot Spots</i> em um ambiente virtualizado ....	21
Figura 3.1 - Serviços de um Datacenter. ....	22
Figura 3.2 - Certificação Tier IV <i>Constructed Facility</i> ....	26
Figura 3.3 - Apple-I.....	28
Figura 3.4 - Tipos de conexões USB.....	30
Figura 3.5 - Arduino UNOR3 e suas conexões. ....	32
Figura 3.6 - Interface IDE 1.0.5 Arduino (Fonte: Autor,2013).....	33
Figura 3.7 - Arquitetura do Arduino.....	33
Figura 3.8 - <i>Shield Ethernet</i> WIZet W5100. ....	34
Figura 3.9 - Sensor DTH11 ....	35
Figura 3.10 - Transistor do tipo NPN TIP120.....	36
Figura 3.11 - LDR ....	36
Figura 3.12 - Resistor e Potenciômetro ....	38
Figura 3.13 - Fan utilizado no projeto modelo: 60x60x25mm-Berflo. ....	38
Figura 3.14 - Servidor DELL Power Edge. ....	39
Figura 3.15 - Modelo de comunicação OSI e modelo TCP/IP.....	40
Figura 3.16 - Protocolo TCP/IP e suas camadas. ....	41
Figura 4.1 - Fluxograma Geral do Modelo.....	42
Figura 4.2 - Esquemático do Projeto. ....	43
Figura 4.3 - Pinagem do Arduino utilizada ....	43
Figura 4.3 - Representação da <i>Shield Ethernet</i> acoplado ao Arduino. ....	44
Figura 4.4 - Início da montagem do protótipo.....	44
Figura 4.5 - Montagem final do protótipo. ....	45
Figura 4.5 - PWM.....	46
Figura 5.1 - Wireless Sensor.....	47
Figura 5.2 - Representação do protótipo de como poderia ser implementado em um Datacenter.....	48

## **LISTA DE TABELAS**

Tabela 3.1 – Especificações Arduino .....	31
Tabela 5.1 – Velocidade da queda de temperatura. ....	48
Tabela 5.2 – Custo do Modelo.....	49

## **RESUMO**

Nos últimos anos, inúmeras empresas, tem se preocupado com a sustentabilidade e ao mesmo tempo com a rentabilidade. Um grande problema das empresas que utilizam e dependem dos serviços de TI é a falta de um correto planejamento do sistema de climatização. Neste trabalho é apresentado os impactos que a densidade térmica pode trazer ao ambiente de um Datacenter e visa implementar uma forma de gerenciamento de ar eficiente minimizando os pontos de calor, reduzindo os custos operacionais, e as interrupções e falhas de processamento relacionados ao aquecimento, refletindo dessa forma em uma melhora na gestão de TI.

**Palavras Chave: Datacenter. HVAC.CCF. Tier. PUE.Arduino.**

## **ABSTRACT**

In recent years, numerous companies have been concerned about the sustainability and profitability at the same time. A major problem of the companies that use and depend on IT services is the lack of a proper planning of the HVAC system. This work impacts the thermal density can bring to a Datacenter environment and aims to implement an efficient way of managing air minimizing hot spots, reducing operational costs, and processing interruptions and failures related to heating, reflecting appears thus in improved IT management.

**Keywords: Datacenter. HVAC.CCF. Tier. PUE.Arduino**



## **CAPÍTULO 1 - INTRODUÇÃO**

O crescimento e a expansão dos negócios das empresas; preocupações como o consumo de energia e impactos ambientais estão assumindo um novo significado.

Datacenters corporativos estão sempre procurando maneiras de ampliar sua capacidade e com isso a demanda por energia para sustentar equipamentos cada vez mais robustos aumentam e consequentemente o resfriamento também precisa se adequar.

### **1.1 – Apresentação do Problema**

Cada vez mais observamos que os custos de energia estão subindo e os equipamentos de Tecnologia da Informação (TI) forçando a infraestrutura de resfriamento. Como podemos contribuir para tratar a resiliência operacional de forma eficiente?

### **1.2 – Objetivos do Trabalho**

**1.2.1– Objetivo Geral:** Desenvolver e implementar, solução que permita a coleta, disponibilidade, análise de dados e controle

**1.2.2 – Objetivos Específicos:**

- Regular a velocidade das ventoinhas que fazem parte do sistema de arrefecimento dos Datacenters.
- Aferição de dados relacionados à temperatura.
- Enviar dados por meio de um Servidor Web.
- Simulação de aquecimento do ambiente onde os servidores são alocados.

### **1.3– Justificativa e Importância do Trabalho**

Um dos grandes desafios encontrado nos Datacenters é o de manter o equilíbrio entre eficiência energética e a operacional. Existem muitas variáveis que devemos considerar mas os resultados que podemos alcançar com a integração de diversas técnicas fazem uma grande diferença. O trabalho em questão vem com a proposta de demonstrar uma das várias facetas encontradas na construção de um Datacenter Verde.

Os calores gerados por equipamentos, cada vez mais robustos, forçam a infraestrutura de resfriamento. Em muitos casos o consumo de energia chega a ser maior com o resfriamento do que a quantidade de energia demandada pelo equipamento.

### **1.4 – Escopo do Trabalho**

O trabalho limita-se a demonstrar uma das várias formas de como podemos trabalhar a densidade térmica gerada pelos servidores utilizando um sistema de resfriamento eficiente e

adequado que não comprometa a performance do equipamento, mantendo assim a estabilidade operacional e a produtividade da corporação.

O referido trabalho não contempla outros métodos de resfriamento e nem outras tecnologias utilizadas nos Datacenters Verdes.

### **1.5 – Resultados Esperados**

- Diminuição da temperatura nos pontos de concentração de calor.
- Melhoramento da eficiência energética.
- Disponibilização dos dados através do Web Server.

### **1.6 – Estrutura do Trabalho**

Esta monografia está estruturada em 6 capítulos iniciando-se com a Introdução no Capítulo 1 que apresenta de forma geral o tema proposto expondo o problema a ser solucionado, objetivos a serem atingidos bem como os resultados esperados.

No Capítulo 2 é apresentado o problema de forma mais aprofundada observando o comportamento dos Datacenters atuais com relação à distribuição de energia; delimitando através de cenários a problemática abordada na presente monografia.

Toda a base metodológica está apoiada no Capítulo 3 que irá abordar e alinhar aspectos teóricos e dispositivos que tornarão possível a resolução do problema

O capítulo 4 é junção da base metodológica com a engenharia de automação e controle resultando em um modelo que irá solucionar o problema observado no Capítulo 2. Inclui descrições técnicas de recursos utilizados e a montagem do protótipo.

No Capítulo 5 é abordado os resultados alcançados envolvendo todos os aspectos desde a apresentação do protótipo e viabilidade até o custo do modelo proposto e problemas enfrentados na implementação.

Por fim, o Capítulo 6 apresenta a conclusão do trabalho como um todo, observando se o resultado obtido atingiu um nível satisfatório com todas as vantagens e limitações do modelo proposto. Melhorias para trabalhos, projetos futuros e refinamento do modelo também fazem parte desse capítulo.

## CAPÍTULO 2 – APRESENTAÇÃO DO PROBLEMA

A eficiência de um Datacenter era medida basicamente através de indicadores vinculados à disponibilidade e ao desempenho. Com o passar do tempo e os aspectos ambientais sendo cada vez mais considerados e o aumento drástico dos custos com energia, os CIOs (*Chief Information Officer* - CIO ou Diretor de Tecnologia da Informação) das empresas estão sendo desafiados e obrigados a repensar melhores estratégias além das já existentes como: confiabilidade, disponibilidade, escalabilidade, redundância e performance de seus Datacenters. O cenário atual requer atenção especial principalmente quando entra-se no aspecto da eficiência energética alinhada ao custo operacional para que o crescimento e expansão dos negócios não sejam afetados.

O Insumo básico desta nova organização, amparada por redes interorganizacionais, passou a ser a informação. A informação, que pode ser conceituada como o dado de significância e valor, é um insumo abundante. Do ponto de vista do sistema econômico vigente, permite, por exemplo, uma menor dependência de insumos escassos como a energia.

Equipamentos de TI de alta densidade já são uma realidade e observa-se que tais equipamentos cada vez mais sobrecarregaram a capacidade de refrigeração e alimentação nos Datacenters. Segundo (Marin, 2011) praticamente 100% da energia elétrica usada no Datacenter é convertida em calor que precisa ser retirado do ambiente de alguma maneira.

Uma das maiores fontes de gastos dentro de um ambiente de Datacenter é a refrigeração. Os projetos de Datacenters em sua grande maioria não procuram otimizar esse aspecto com a intenção de economizar, o que é um grande erro. Atualmente existem várias técnicas algumas mais simples outras mais complexas mas que podem otimizar a refrigeração de um Datacenter reduzindo custos e diminuindo os problemas relacionados à manutenção.

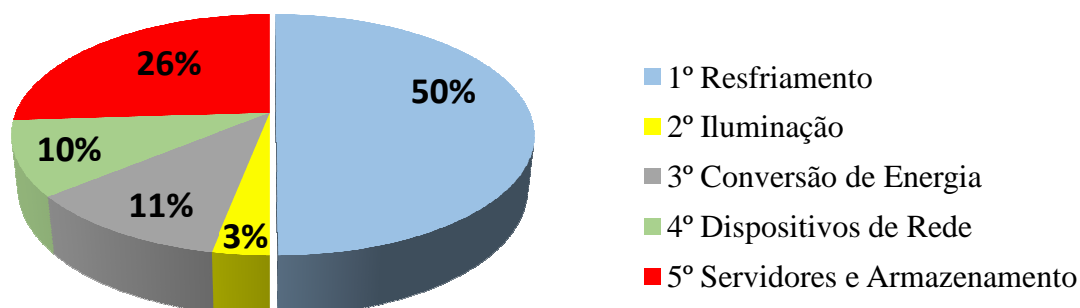


Figura 2.1 - Consumo Típico de Energia em um Datacenter (Fonte: *Info-Tech Research Group*, 2011 adaptado).

O *New York Times* publicou um artigo em setembro de 2012 chamando a atenção para a enorme quantidade de energia consumida pelos Datacenters. As enormes instalações, que supostamente são mais de 3 milhões em todo o mundo, são responsáveis por mais de 2% do consumo mundial de eletricidade. Uma parcela significativa dessa energia é usada para resfriar os equipamentos a fim de evitar o superaquecimento e falhas como dito anteriormente.

A figura 2.1 ilustra e demonstra que de 100% da energia injetada no Datacenter, só 36% foi utilizada efetivamente pela carga de TI, 11% foi consumida pelos equipamentos que transformam e adequam a energia que chega à carga de TI e 50% foi consumida pelos equipamentos de refrigeração e 3% com a iluminação desses ambientes.

O projeto de climatização ou sistema HVAC (*Heat, Ventilation and Air Conditioning*; Aquecimento, Ventilação e Condicionamento de Ar) é tão importante e crítico quanto o projeto do sistema de alimentação e distribuição de energia elétrica. (MARIN, 2011)

O aumento da densidade de potência no *rack* (armário ou gabinete, que abriga por padrão através de normas técnicas todo o material associado à rede local do edifício) faz com que o ambiente sofra uma mudança também na densidade de calor. Um sistema de refrigeração precisa estar preparado para distribuir a refrigeração nas dimensões corretas que é crucial para identificar e explorar os potenciais de poupança de energia.

O PUE (*Power Utilization Effectiveness*, Eficiência da Utilização da Energia elétrica) é uma medida do desempenho global de um Datacenter. O PUE mede o número de watts necessários para alimentar e arrefecer os equipamentos de TI, tendo em conta toda a energia dedicada unicamente ao Datacenter.

$$PUE = \frac{C_{infraestrutura}}{C_{TI}}$$

$C_{infraestrutura}$  sendo a carga total de infraestrutura de em kW; e  $C_{TI}$  a carga total dos equipamentos de TI instalados no Datacenter, em kW. O PUE cujo o valor é 1,00 está associado à uma eficiência de 100%, ou seja, toda a carga disponível para a alimentação dos equipamentos do Datacenter são consumidas pelos equipamentos de TI.

O CCF (*Cooling Capacity Factor*, Fator de Capacidade de Refrigeração) é outra métrica simples e prática para avaliar, calcular e interpretar dados relativos à refrigeração em ambientes de Datacenters. O CCF é calculado dividindo-se a capacidade total de resfriamento em execução do equipamento em kW declarado pelo fabricante por 110% da carga crítica de TI.

A capacidade total de resfriamento em execução é a soma das unidades de refrigeração que funcionando nas capacidades nominais. Se todas as unidades de refrigeração estão em execução, então este será o mesmo valor que a capacidade total instalada de refrigeração nominal. Por exemplo, se existem 10 unidades de refrigeração instalados com uma capacidade nominal de 30 toneladas cada, e sete estão em execução, em seguida, a execução total capacidade de refrigeração é 739 kW (7 x 30 toneladas = 210 toneladas, 210 toneladas x 3,52 = 739 kW). Para converter toneladas de kW, multiplique toneladas pela constante 3,52.

Para a conta da carga da *computer room* não refletir na saída da UPS (*Uninterruptible Power Supply*, Fonte de Energia Ininterrupta) adiciona-se 10% para luzes, pessoas e sala.

$$CCF = \frac{\text{capacidade total de resfriamento em execução}}{\text{Saída UPS} \times 1.1}$$

A *Upsite Technologies* é uma empresa norte americana pioneira em soluções de gerenciamento de fluxo de ar em Datacenters e avaliou 45 localidades e coletou dados com relação a capacidade de refrigeração desses ambientes e a média foi surpreendentemente 3,9 maior que a carga de calor gerado no Datacenter, ou seja, 390% maior do que era necessário para manter a refrigeração em níveis aceitáveis.

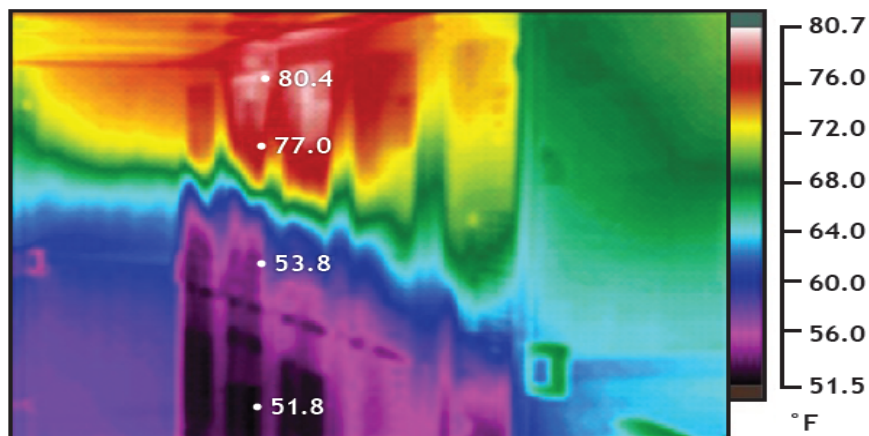


Figura 2.2 - Temperatura dentro de um dos 45 Datacenters analisados (Fonte: *Upsite Technologies*, 2013).

A Figura 2.2 ilustra uma diferença dramática de temperatura na entrada de equipamentos de TI localizados na parte inferior dos *racks* contra os localizados no topo utilizando um corredor do tipo *fully-contained-cold-aisle* que seria em teoria um corredor frio totalmente isolado. Este é o resultado de um volume insuficiente de ar condicionado que está sendo entregue ao espaço contido devido às aberturas de cabo não seladas em todo o piso elevado e outros problemas.

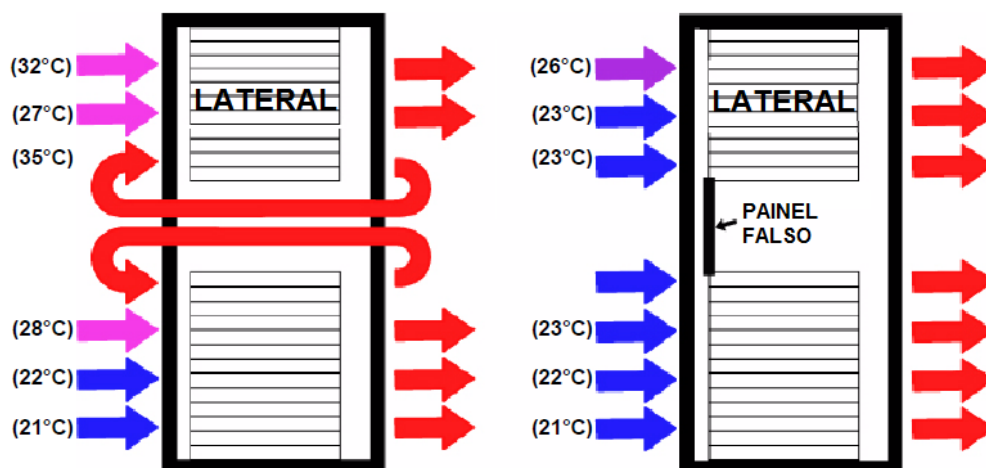


Figura 2.3 - Temperatura dentro de um Datacenter (Fonte: *Schneider Electric*, adaptado).

Apesar do método de corredor contido utilizado na figura 2.3, o comportamento do fluxo de ar com e sem painéis falsos entre os espaços nos equipamentos de TI no rack. Sem o painel falso permite-se que o ar condicionado flua fora do espaço contido e libere o ar que deveria ficar no corredor quente. Mesmo com a instalação do corredor frio de contenção o local apresentou um CCF de 3,8 (380%). A solução implementada da forma errada compromete toda a infraestrutura de resfriamento. Outras soluções também podem melhorar significativamente a infraestrutura de resfriamento como:

- Vedação de aberturas de cabos.
- Ajuste no número de placas perfuradas e localização.
- Ajuste na posição dos *Racks*.
- Selar espaços onde não há mais *Racks*.

Quando se tem a capacidade de refrigeração excessivamente implementada do que é requerido pelo Datacenter, pode-se então, explorar nesse ambiente, potenciais reduções no custo operacional. Desligar unidades de resfriamento e/ou reduzir a velocidade das *fans* que são as ventoinhas de velocidade variável ou utilizar as (VFD *Variable Frequency Drives*, Unidades de Frequência Variável), são exemplos também dessas técnicas que reduzem a média da despesa operacional em um Datacenter, que segundo observado pelo *Upsite Technologies* chega à atingir aproximadamente \$32.000,00 por ano apenas com o (AFM *Airflow Manegement*, Gerenciamento do fluxo de Ar).

Através do gerenciamento do fluxo de ar (AFM) pode-se aumentar a eficiência de arrefecimento o que resulta em economia imediata com os custos operacionais refletindo em uma maior confiabilidade do sistema de TI, pode também, liberar capacidades presas de resfriamento permitindo às empresas aumentarem a densidade dos servidores sem o custo

adicional de capital com equipamentos de refrigeração. A utilização de refrigeração melhorada também pode estender a vida útil de um Datacenter adiando os investimentos necessários para a construção de um novo Datacenter.

Em maio de 2011, a ASHRAE (*American Society of Heating, Refrigeration, and Air conditioning Engineers*, Sociedade Americana de Engenheiros de Climatização), entidade norte-americana de renome internacional na área de padronização para a climatização, publicou diretrizes atualizadas para o que é "recomendado" e "admissível" dentro dos Datacenters. Com a nova especificação criaram-se duas novas classificações de Datacenters e ampliaram-se o leque de condições ambientais permitidas, a fim de incentivar práticas de eficiência energética, como o uso de ar externo (*freecooling*). *The Green Grid* publicou mapas de arrefecimento livre em 2009, baseado na *ASHRAE* 2008 com intervalos de temperatura e umidade recomendados. Agora, com as faixas admissíveis mais amplas, novos mapas mostrando o potencial de expansão para o resfriamento com o uso de ar externo são necessários para incentivar as equipes de projeto à incorporar essas técnicas de poupar de energia.

Com o desenvolvimento de processadores mais rápidos e melhores e o desenvolvimento continuado da amplitude de banda das fibras que são responsáveis pelo transporte de dados, começa a existir uma necessidade crescente de fluxo de ar.

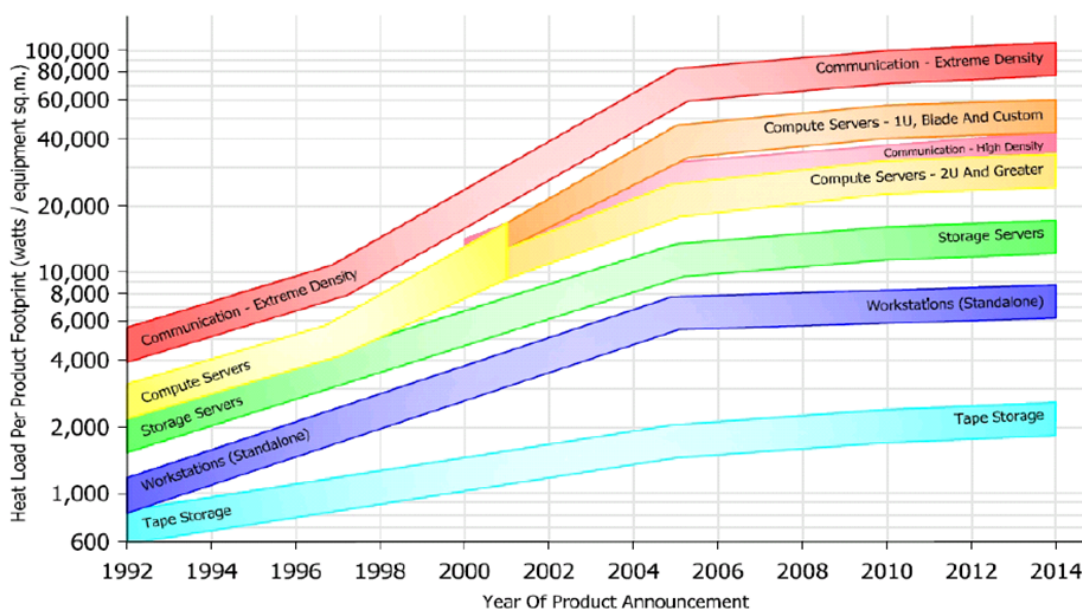


Figura 2.4 - Evolução de carga térmica (Fonte:ASHRAE).

A figura 2.4 ilustra a tendência para a carga de refrigeração de diferentes sistemas de TI (expressas em Watts/m<sup>2</sup>) para os próximos anos. Pode notar-se que a potência específica para o resfriamento instalado vai continuar aumentando. Assim, adotar estratégias de sistemas



mais eficientes juntamente com melhores práticas e gestão de sistemas de refrigeração são importantes. A eficiência energética nos Datacenters e salas de infraestruturas de servidores tornou-se um grande problema para a TI.

O CFD (*Computational Fluid Dynamics*, Dinâmica de Fluidos Computacionais) é uma simulação numérica de escoamento de fluido. Pode ser utilizada para prever velocidades de fluidos, temperaturas e muitas outras variáveis de interesse para a grande variedade de aplicações. O objetivo desse cálculo é prever o campo de fluxo altamente detalhado em alguma região de interesse. Atualmente existem softwares de simulação de fluidos computacionais que conseguem prever o fenômeno físico termo/líquido no Datacenter.

Com uma análise 3-D detalhada de como o ar frio se move através de um Datacenter, a identificação de potenciais *hot spots* ou pontos quentes em que o equipamento está recebendo um fluxo de ar muito pequeno. Utilizando também o mapeamento térmico é possível encontrar áreas em um Datacenter que estão recebendo mais ar frio do que o necessário, desperdiçando assim refrigeração e energia. A dinâmica dos fluidos computacionais também é útil para modelar o fluxo de ar sob um piso elevado, o que é bastante interessante quando se inicia um projeto de refrigeração em Datacenter. Na figura 2.5 observa-se o comportamento dos fluidos.

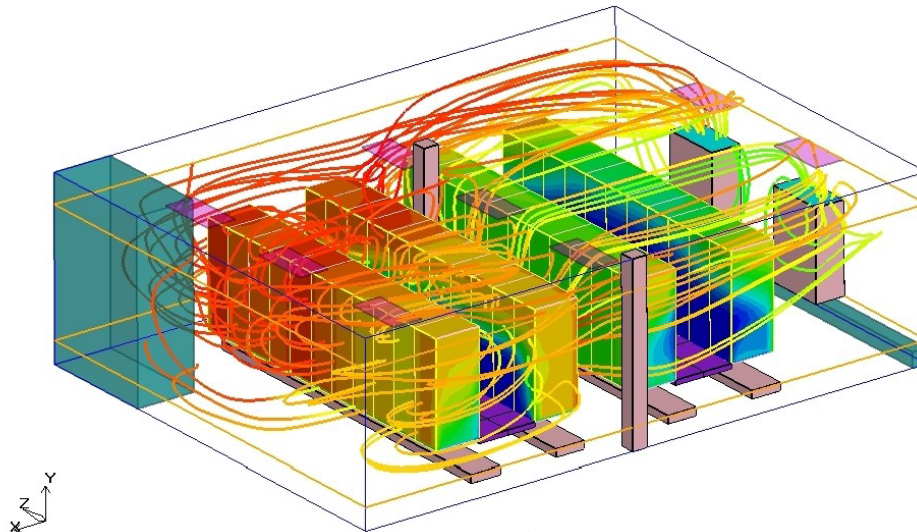


Figura 2.5 - Um modelo CFD em um Datacenter (Fonte: *Applied Math Modeling Inc.*).

A virtualização pode ser definida como a criação de um ambiente virtual que simula um ambiente real, propiciando a utilização de diversos sistemas e aplicativos sem a necessidade de acesso físico à máquina na qual estão hospedados.

A consolidação da virtualização é uma das principais opções para aumentar significativamente a eficiência energética em Datacenters. No entanto, quando inicia-se um



projeto de refrigeração, a poupança completa só pode ser atingida se a infra-estrutura, incluindo o abastecimento de energia e refrigerador, for tratada simultaneamente.

Ao reduzir significativamente a procura de energia global, a virtualização, especialmente em sistemas maiores, é evidente o aumento da densidade de potência por *rack*.(Figura 2.6).

Além disso, a gestão de energia pela migração de máquinas virtuais leva a uma mudança dinâmica espacial da energia e da densidade de calor, portanto, aumenta localmente a procura de energia e refrigeração. Os conceitos apropriados de alimentação e de refrigeração precisam ser usados para atender às necessidade de ambientes virtualizados e evitar *hot spots* ou pontos de excessiva concentração de calor

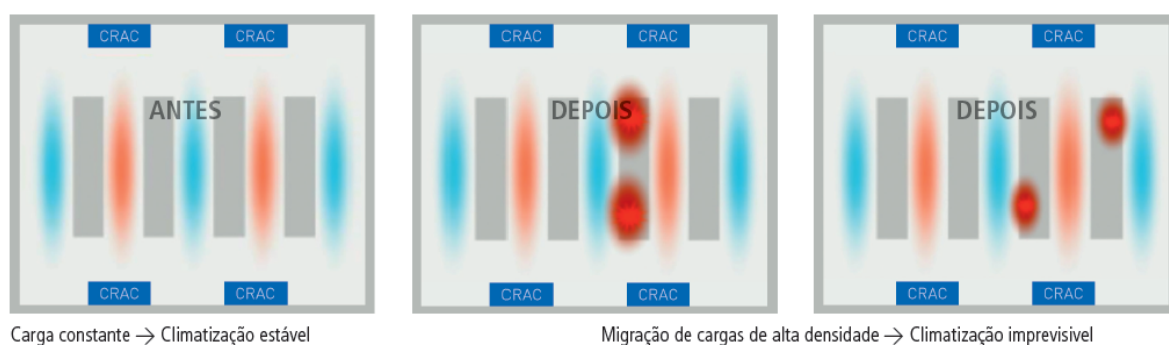


Figura 2.6 - *Hot Spots* em um ambiente virtualizado (Fonte: *Prime Energy IT*).

No referido trabalho como explanado no item 1.4 será abordado principalmente os problemas encontrados e soluções já existentes relacionados aos: tipos, modelos, tecnologias e soluções de refrigeração utilizadas nesses ambientes.

## CAPÍTULO 3 – BASES METODOLÓGICAS PARA RESOLUÇÃO DO PROBLEMA

### 3.1 – Visões da Infraestrutura de TI

Reagindo ao aumento crescente da concorrência inúmeras empresas têm buscado a utilização de um formato organizacional mais adaptável. Diferentemente da verticalização, que foi um modelo utilizado há décadas atrás, onde todo o processo organizacional era executado dentro de uma única empresa, a ampla utilização de redes que interligam organizações, contribuindo de forma a apoiar os processos, é uma característica e tendência que permitem que as organizações atuem em conjunto representando um grande sistema agregador de valor.

A infraestrutura de TI, como qualquer outra infraestrutura, tem o papel de possibilitar que a organização possa funcionar e crescer sem grandes interrupções. Na prática, do ponto de vista do usuário dos processos de negócio o que interessa é a qualidade do serviço que ele está recebendo proveniente da equipe de infraestrutura de TI, ou seja, o usuário de TI olha para a TI do ponto de vista de tomador de decisões. (VERAS, 2010).

#### 3.1.2 – Visão Geral do Datacenter

Um Datacenter é classificado, primordialmente, por suas características de disponibilidade, confiabilidade e redundância conforme figura 3.1. Esses ambientes devem estar aptos a operar em níveis de disponibilidade perto de 100% (ver item 3.1.2.3) garantindo assim uma operação ininterrupta.(VERAS, 2010).

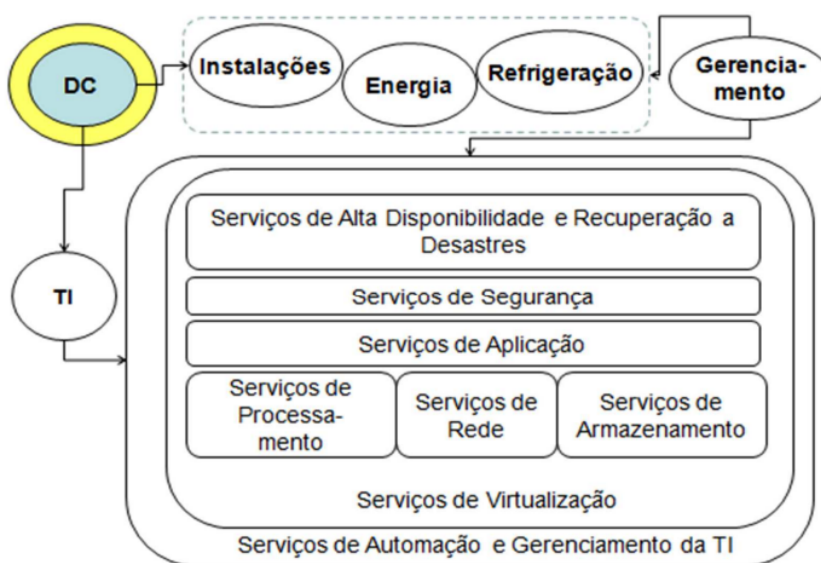


Figura 3.1 - Serviços de um Datacenter (Fonte: VERAS, 2010).

### 3.1.2.1 – Conceito

Datacenters são conhecidos como ambientes de missão crítica, ou seja, ambientes que abrigam equipamentos responsáveis pelo processamento e armazenamento de informações cruciais para a continuidade do negócio em vários tipos de organizações, sejam empresas, instituições de ensino, órgãos governamentais, hospitais dentre outros. (MARIN, 2011).

Embora os projetos de Datacenters sejam únicos, eles geralmente podem ser classificados como Internet ou *Enterprise* (ou "interno"). Datacenters voltados para a Internet suportam relativamente poucos aplicativos, são tipicamente baseados em browser, e se caracterizam por suportar muitos usuários, geralmente desconhecidos. Por outro lado, um Datacenter do tipo *Enterprise* suportam menos usuários, mas fazem o uso de aplicações que variam de básicas para aplicações personalizadas.

Arquiteturas de Datacenters e os requisitos podem variar significativamente. Por exemplo, um Datacenter construído para um prover serviços na nuvem como o Amazon EC2 satisfazem facilidades significativamente diferentes de infraestrutura e requisitos de segurança, diferentemente de um Datacenter totalmente privado como um construído para o Pentágono, que se dedica a manter em segurança dados sigilosos.

Independentemente da classificação, um eficaz funcionamento do um Datacenter é alcançado através de um investimento equilibrado nas instalações e equipamentos. Os elementos de um Datacenter podem ser quebrados em:

Facilidade: a localização, "espaço em branco" ou espaço utilizável, que está disponível para equipamentos de TI. Proporcionando o acesso permanente à informação fazem com que os Datacenters sejam uma das instalações mais consumidoras de energia do mundo. A grande ênfase é colocada sobre design para otimizar espaços em branco e controle ambiental para manter o equipamento dentro do especificado pelo fabricante que são: temperatura e umidade.

Infraestrutura de apoio: equipamento que contribui para sustentar firmemente o mais alto nível de disponibilidade possível. O *Uptime Institute* definiu quatro níveis (*Tiers*) de Datacenters que será explicado mais detalhadamente no item 3.1.2.3 onde a disponibilidade varia de 99,671 % à 99,995%. Alguns componentes para apoiar a infraestrutura incluem:

- UPS:(*Uninterruptible Power Sources*, Fontes de alimentação ininterrupta), bancos de baterias, geradores e fontes de alimentação redundantes.
- Controle do ambiente: CRAC (*Computer Room Air Conditioners*, Ar condicionado da Sala de Computador),HVAC (*Heat, Ventilationand Air Conditioning*, Aquecimento, Ventilação e Condicionamento de Ar) e sistemas de exaustão.

- Sistemas de segurança física: biometria e sistemas de vigilância por vídeo.

Equipamentos de TI: equipamentos para as operações de TI e de armazenamento de dados da organização. Isso inclui servidores, equipamentos de armazenamento, switches e roteadores, cabos e *racks*, bem como uma variedade de elementos de segurança da informação, tais como firewalls.

Equipe de operações: para monitorar as operações e manutenção de TI e equipamentos de infraestrutura.

Os Datacenters têm evoluído significativamente nos últimos anos, a adoção de tecnologias como a virtualização para otimizar a utilização dos recursos de TI e aumentar a flexibilidade. Como empresa de TI precisam continuar a evoluir para serviços sob demanda, muitas organizações estão se movendo em direção a serviços e infraestrutura baseados na nuvem (*cloud computing*).

### 3.1.2.2 – Histórico

Os primeiros computadores eram complexos para operar e manter, e exigiu um ambiente especial para o seu funcionamento. Muitos cabos eram necessários para conectar todos os componentes e métodos para acomodar e organizar estes computadores foram criados, como o padrão de *racks* para montar equipamentos, pisos elevados, e calhas de cabos (instalados por cima ou sob o piso elevado). Além disso, um único mainframes necessitava de uma grande quantidade de energia, e teve que ser resfriado para evitar o superaquecimento.

Durante o boom da indústria de microcomputadores, e especialmente durante a década de 1980, os computadores passaram a ser implantados em todos os lugares, em muitos casos com pouca ou nenhuma atenção sobre os requisitos operacionais. No entanto, como as operações de tecnologia da informação (TI) começaram a crescer em complexidade, as empresas cresceram conscientes da necessidade de controlar os recursos de TI.

Com o advento do Linux e consequentemente proliferação de sistemas operacionais Unix disponíveis gratuitamente e compatíveis com PC durante a década de 1990, bem como o MS-DOS que deu caminho ao sistema operacional multi-tarefa Windows, PCs começaram a encontrar seus lugares nas antigas salas de informática. Estes foram chamados de "servidores". A disponibilidade e baixo custo de rede de equipamentos, juntamente com novos padrões de rede de cabeamento estruturado, possibilitaram o uso de um design hierárquico que colocava os servidores em uma sala específica dentro da empresa. O uso do termo "Datacenter", foi concebido e aplicado às *computer rooms* ou sala de computadores, que começaram a ganhar reconhecimento popular.

O ápice dos Datacenters aconteceu durante a bolha do ".com" . Empresas necessitavam de uma conectividade rápida com a Internet e operação ininterruptas para implantar sistemas e estabelecer uma presença na Internet. O crescimento do poder de processamento de servidores baseados em tecnologia X86 e a opção de utilizar processadores com vários núcleos permitiram fazer o que se convencionou chamar de recentralização ou consolidação trouxe novamente a conceito de Datacenter agora baseado em tecnologias padrões (VERAS, 2010).

### 3.1.2.3 – Classificação de DATACENTERS em *TIERS 1, 2, 3 e 4*

O *Uptime Institute* iniciou um projeto de estudo e classificação de Datacenters em 1995, sendo atualmente largamente utilizado como referência em projetos de Datacenter e tem como objetivo delimitar a topologia para os projetistas e engenheiros bem como mensurar o desempenho final requerido pelo nível de criticidade do projeto.

O *Uptime Institute* também fornece certificações de projetos, e instalações e programas de gestão.

Abaixo um breve resumo e diferenças entre esses níveis.

#### No *Tier1*

- ✓ Caminho único de distribuição não redundante servindo os equipamentos de TI.
- ✓ Componentes e capacidade não redundantes.
- ✓ Infraestrutura básica local garantindo 99,671% de disponibilidade.

#### No *Tier2*

- ✓ Atende ou excede todas as exigências do *Tier1*.
- ✓ Componentes da capacidade de infraestrutura local redundante garantem 99,741% de disponibilidade.

#### No *Tier3*

- ✓ Atende ou excede todos os requisitos *Tier1* e *Tier 2*.
- ✓ Vários caminhos de distribuição que servem os equipamentos de TI
- ✓ Todos os equipamentos de TI devem ser duplamente energizados *dual-powered* e totalmente compatíveis com a topologia da arquitetura de um site.
- ✓ Ao mesmo tempo possuir infraestrutura local sustentável garantindo 99,982% de disponibilidade.

#### No *Tier4*

- ✓ Atende ou excede todos os requisitos do Nível 1, Nível 2 e Nível 3
- ✓ Todos os equipamentos de refrigeração são independentes, ou seja, duplamente

energizados *dual-powered*, incluindo *chillers* e sistemas de aquecimento, ventilação e ar condicionado (HVAC).

- ✓ Tolerante à falhas de infraestrutura local com armazenamento de energia elétrica e instalações de distribuição garantindo 99,995% de disponibilidade.

Em julho de 2013 no Brasil o Banco Santander Brasil S/A Centro Tecnológico Campinas – DC1 e DC2 recebeu a Certificação *Tier IV Constructed Facility*.

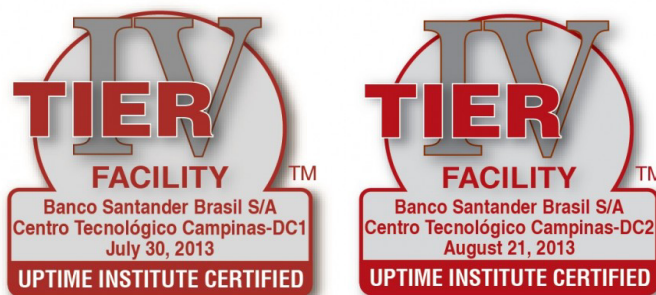


Figura 3.2- Certificação *Tier IV Constructed Facility*.(Fonte: *Uptime Institute*).

#### 3.1.2.4 – Tendências

2014 representa o 50º aniversário da introdução do mainframe 360<sup>6</sup> da IBM.

Como 34% da população do mundo (mais de 2,5 bilhões de pessoas) desfruta de acesso à Internet, e com as empresas contando com comunicações baseadas em IP (voz e vídeo), bem como serviços em nuvem, as demandas que estão sendo colocadas em centros de dados de hoje são imensas.

Muitos Datacenters que existem hoje, e suportam os serviços de missão crítica, tanto nos ambientes "comercial" e "doméstico" são baseados em tecnologias de 20 anos de idade, e o simples fato é que eles não podem mais acompanhar a demanda. Os analistas prevêem que o número de servidores vai aumentar em dez vezes nos próximos três anos, e que 70% de todo o tráfego de Datacenter em 2015 será entre os servidores.

### 3.2 – DATACENTER Verde (*Green DATACENTER*)

Um foco também foi colocado em iniciativas para reduzir o enorme consumo de energia dos centros de dados através da incorporação de tecnologias e práticas mais eficientes na gestão de Datacenters para minimizar o impacto ambiental. Datacenters construídos com essas normas são comumente conhecidos como *Green Datacenters*.

Segundo Merin (2011), Independente do tamanho do Datacenter, seu consumo de energia elétrica será elevado dada a grande quantidade de equipamentos de TI instalados nesses ambientes. Servidores *blade* (servidores em um único chassi), equipamentos de *storage*

(equipamentos para armazenamento de dados), redes SAN (*Storage Network Area*, Rede de Área de Armazenamento), NAS (*Network-Attached Storage*, Armazenamento conectado à Rede) dentre outros são exemplos de equipamentos que consomem um elevado consumo de energia elétrica.

Outros sistemas dentro do Datacenter como os equipamentos de climatização de precisão também são responsáveis pelo excessivo consumo de energia elétrica.

### **3.3 – Hardware e Interfaces para Comunicação com o Circuito**

#### **3.3.1 – Computador Pessoal**

O PC (*Personal Computer*; Computador Pessoal) é um computador de uso geral, com tamanho, capacidade, preço acessível e de grande utilidade para indivíduos, que se destina a ser operado diretamente por um usuário final sem qualquer intervenção de outro operador.

Os primeiros computadores pessoais geralmente chamados de microcomputadores foram vendidos em forma de kits e em volumes limitados, eram de interesse principalmente para amadores e técnicos. Uma programação básica era feita com interruptores para a entrada de instruções e a saída fornecida por lâmpadas em um painel frontal. Para o uso prático era necessário a adição de periféricos como teclados, monitores de computador, discos rígidos e impressoras.

O *Micral N* foi o microcomputador comercial baseado em um microprocessador, o Intel 8008. Foi construído a partir de 1972 e cerca de 90 mil unidades foram vendidas. Em 1976, Steve Jobs e Steve Wozniak venderam o computador *Apple-I* construído em uma placa de circuito, que foi totalmente preparada e continha cerca de 30 chips. O primeiro computador pessoal com sucesso comercializados em massa foi o *Commodore PET* introduzido em Janeiro de 1977, mas por conta da pouca produção não estava mais disponível até o final do ano. Foi logo seguido pelo *Apple II* (geralmente referida como a "*Apple*") em junho de 1977, e o *TRS-80* da *Radio Shack*, em novembro de 1977. Com a produção de computadores em larga escala e já montados foi possível uma maior parcela de pessoas que pudessem utilizá-los. O mercado se concentrou então no desenvolvimento de software e menos no desenvolvimento do hardware do processador.



Figura 3.3 - Apple-I (Fonte: [apple2history.org](http://apple2history.org))

Durante o início da década de 1980, os computadores domésticos foram desenvolvidos para uso doméstico, com software de produtividade pessoal, programação e jogos. Eles normalmente poderiam ser usados com uma televisão em casa utilizando o monitor do computador, com gráficos em blocos e poucos detalhes e uma gama de cores limitada, e um texto de cerca de 40 caracteres de largura por 25 caracteres de altura. Um tipo de máquina, o *ZX Spectrum* introduzido em 1982, totalizou 8.000.000 unidades vendidas. A seguir veio o *Commodore 64*, que totalizou 17 milhões de unidades vendidas, tornando-se o modelo de computador pessoal mais vendido de todos os tempos. Outro exemplo de computador, o *NEC PC-98*, vendeu mais de 18 milhões de unidades.

As estações de trabalho foram caracterizadas por processadores de alto desempenho e monitores gráficos, com grande armazenamento em disco local, capacidade de trabalho em rede, e que funcionam sob um sistema operacional multitarefa.

Finalmente, devido à influência da IBM PC no mercado não havia mais qualquer distinção técnica entre computadores pessoais, computadores e computadores domésticos. Computadores de negócio foram adquirindo capacidade gráfica cor e som, e computadores domésticos e usuários de sistemas de jogos utilizavam os mesmos processadores e sistemas operacionais como os trabalhadores de escritório. Computadores em massa do mercado tinham capacidades gráficas e memória comparáveis às estações de trabalho dedicadas de alguns anos antes. Mesmo rede de área local, originalmente uma forma de permitir que os computadores de negócios para compartilhar de armazenamento em massa caro e periféricos, tornou-se uma característica padrão dos computadores pessoais usados em casa.

Em 1982, "O computador" foi chamado de Máquina do Ano pela Time Magazine .



### 3.3.2 – Porta USB

O USB(*Universal Serial Bus*; Barramento Serial Universal) é um padrão da indústria, desenvolvido em meados da década de 1990 que define os cabos, conectores e protocolos de comunicação usados em um barramento para a conexão, comunicação e fonte de alimentação entre computadores e dispositivos eletrônicos.

Em 1994 um grupo de sete empresas: Compaq, DEC, IBM, Intel, Microsoft, NEC e Nortel. Começaram o desenvolvimento do USB(*Universal Serial Bus*; Barramento Serial Universal). O objetivo era torná-lo fundamentalmente mais fácil de conectar dispositivos externos para PCs, substituindo a infinidade de conectores na parte de trás de PCs, abordando as questões de usabilidade de interfaces existentes, e simplificando a configuração de softwares todos os dispositivos conectados ao USB, bem como permitindo maior taxas de dados para dispositivos externos. A equipe, incluindo AjayBhatt (Arquiteto indiano) trabalhou no padrão da Intel; os primeiros circuitos integrados de suporte USB foram produzidos pela Intel em 1995.

A especificação original USB 1.0, que foi lançado em Janeiro de 1996, definiu a taxa de transferência de dados em 1.5 Mbit/s como "*LowSpeed*" e 12 Mbit/s "*FullSpeed*". A primeira versão amplamente utilizada do USB foi de 1.1, que foi lançado em setembro de 1998. A taxa de transferência de dados de 12 Mbit/s foi destinada para dispositivos de maior velocidade, tais como unidades de disco, e a taxa de transferência de dados de 1,5 Mbit/s para dispositivos de menor velocidade como joysticks.

A especificação USB 2.0 foi lançado em abril de 2000 e foi ratificada pelo *USB Implementers Forum* (USB-IF), no final de 2001. A HP (*Hewlett-Packard*), Intel, Lucent *Technologies* (atualmente Alcatel-Lucent), NEC e Philips planejaram em conjunto a iniciativa de desenvolver taxas de transferência de dados mais altas, com a especificação resultante atingindo 480 Mbit/s, um aumento de 40 vezes a especificação original do USB 1.1.

A especificação USB 3.0 foi publicado em 12 de novembro de 2008. Seus principais objetivos foram: aumentar a taxa de transferência de dados chegando à aproximadamente 5 Mbit/s, diminuir o consumo de energia, aumento de energia na saída e ser compatível com o USB 2.0. O USB 3.0 inclui um novo barramento de maior velocidade chamado *SuperSpeed* em paralelo com o barramento USB 2.0. Por esta razão, a nova versão é também chamado *SuperSpeed*. Os primeiros dispositivos equipados com USB 3.0 foram apresentados em Janeiro de 2010.

Em setembro de 2013 a *Fresco Logic* uma empresa de semicondutores global que desenvolve e comercializa soluções de conectividade avançada, demonstrou a transferência de

dados do *SuperSpeed* USB 10 Gbps (USB 3.1) em sua plataforma de desenvolvimento de hardware no *Intel DeveloperForum* (IDF). A taxa de transferência de dados alcançada foi de 10 Gbps em um link bidirecional, mais que o dobro da largura de banda efetiva da já rápida USB (USB 3.0) padrão.

O USB foi projetado de modo que possam ser ligados vários periféricos pelo mesmo canal (i.e., porta USB). Assim, mediante uma topologia em árvore, é possível ligar até 127 dispositivos a uma única porta do computador, utilizando hubs especialmente criados para essa função. (MORAZ, 2006)

Todos os dispositivos USB têm uma hierarquia que informam ao Host o que o dispositivo é, ou seja, suas características de funcionamento, como o número de série do produto, identificação do fabricante, tipo do dispositivo (impressora, scanner, modem, mouse, etc.), número de configurações, número de *Endpoint*, tipo de transferência, tipo de interface, etc. Na figura 3.4 observa-se os tipos de conexões USB.

Os dispositivos que podem se conectar à porta USB, precisam de um driver específico para serem usados ou podem pertencer a uma classe de dispositivo. Estas classes definem um comportamento já esperado de determinados dispositivos, sendo assim, pode ser utilizado um mesmo driver para qualquer equipamento que pertença a mesma classe. Um sistema operacional geralmente possui drivers genéricos para todas as classes.

Transações USB consistem de três pacotes:

- Token
- Data
- Status

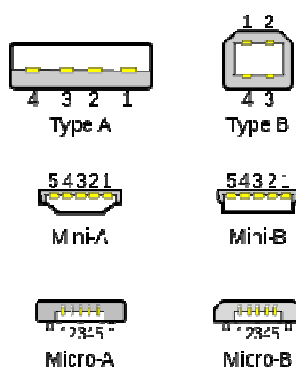


Figura 3.4 - Tipos de conexões USB (Fonte: [www.sproboticworks.com](http://www.sproboticworks.com)).

Há alguns anos atrás para que fosse possível montar um circuito com uma complexidade maior era necessário usar inúmeros componentes como: CI (Circuitos Integrados), capacitores, resistores, transistores e outros. Tal circuito após montado era especificamente destinado a realizar uma determinada tarefa e as correções necessárias após montado em uma placa de circuito impresso era um fator complicador.

Os Microcontroladores surgiram para facilitar o projeto de circuitos eletrônicos mais complexos onde trilhas nas placas de circuitos impressos e componentes são substituídos por linhas de códigos inseridos dentre desses microcontroladores. Através da reescrita de algumas linhas do código de um determinado programa pode-se alterar radicalmente todo o funcionamento de um circuito, ao invés de trocar vários componentes em um placa. O conhecimento de uma linguagem de programação passa a ser um elemento importante para os engenheiros eletrônicos para trabalhar com esse tipo de dispositivo.

### 3.5 – O Arduino

O Arduino UNO R3 ("Uno" significa um em italiano e é nomeado para marcar o lançamento do Arduino 1.0) utilizado no projeto é uma placa microcontroladora baseada no ATmega328. Possui 14 pinos para entrada digitais de entrada e saída onde 6 pinos podem ser utilizados como saída PMW (*Pulse-Width Modulation*, Modulação por Largura de Pulso) de 8-bit e uma delas é utilizada no projeto, 6 pinos para entradas analógica, um cristal oscilador de 16Mhz, uma conexão USB, um conector de alimentação, um cabeçalho ICSP (*In Circuit Serial Programming*) e um botão de *reset*. Utiliza código fonte aberto tornando a criação de projetos interessante e de forma relativamente rápida e fácil. o UNO se difere das placas anteriores por não usar o *driver* do *chip FTDI USB-to-serial*; ele utiliza o ATmega16U2 programado com um conversor *USB-to-serial*. (Figura 3.5)

Tabela 3.1 – Especificações Arduino.

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA

DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
ClockSpeed	16 MHz

Fonte: <http://arduino.cc/>

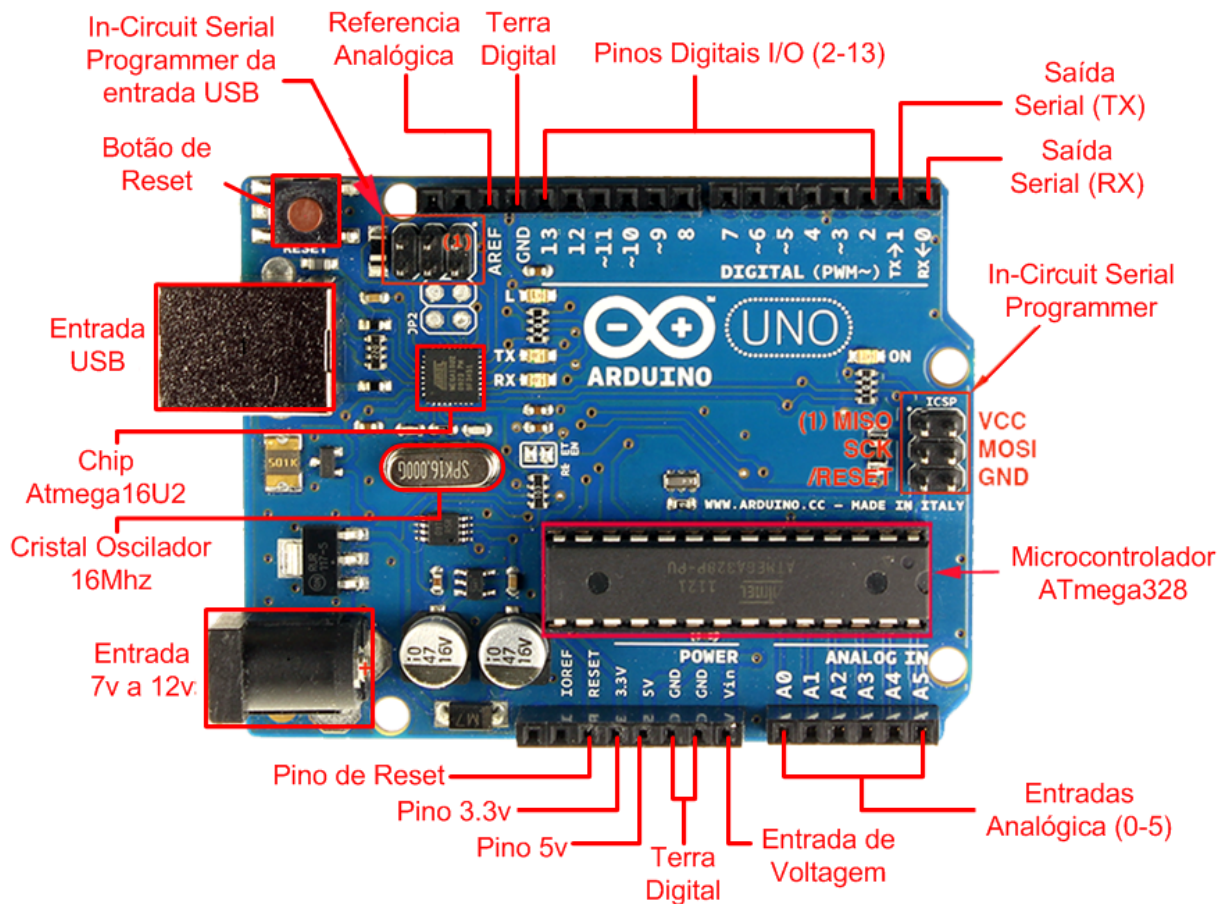


Figura 3.5 - Arduino UNOR3 e suas conexões (Fonte: Autor).

Para programá-lo utiliza-se o IDE(*IntegratedDevelopmentEnvironment*, ou Ambiente de Desenvolvimento Integrado) esse software é utilizado para escrever os códigos. Figura 3.6

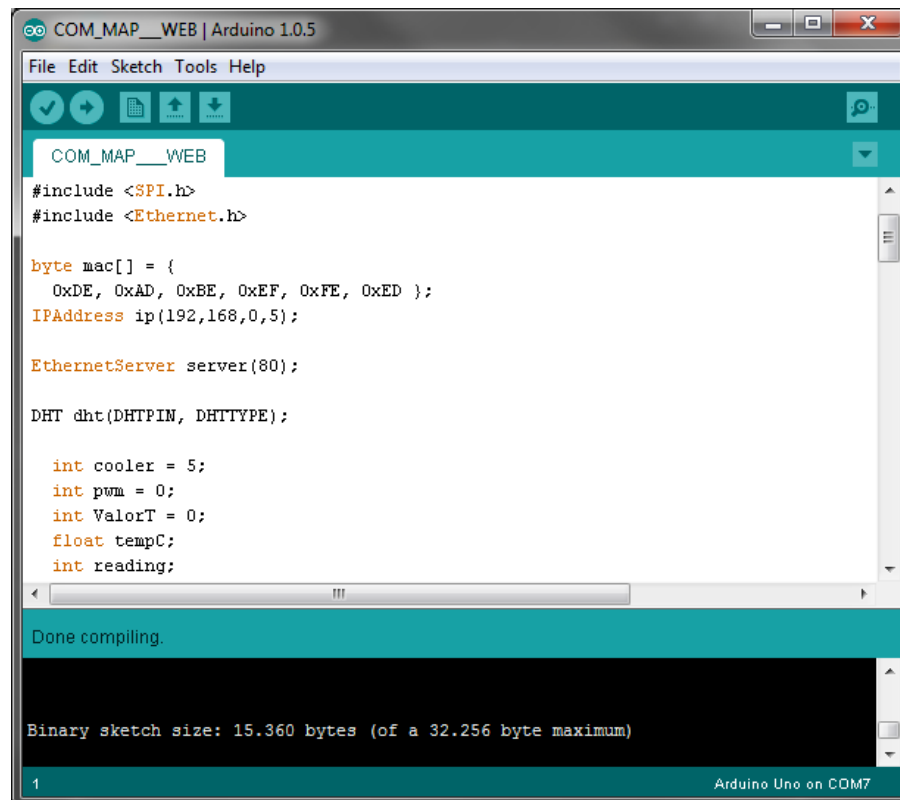


Figura 3.6 - Interface IDE 1.0.5 Arduino (Fonte: Autor,2013).

### 3.5.1 – Arquitetura

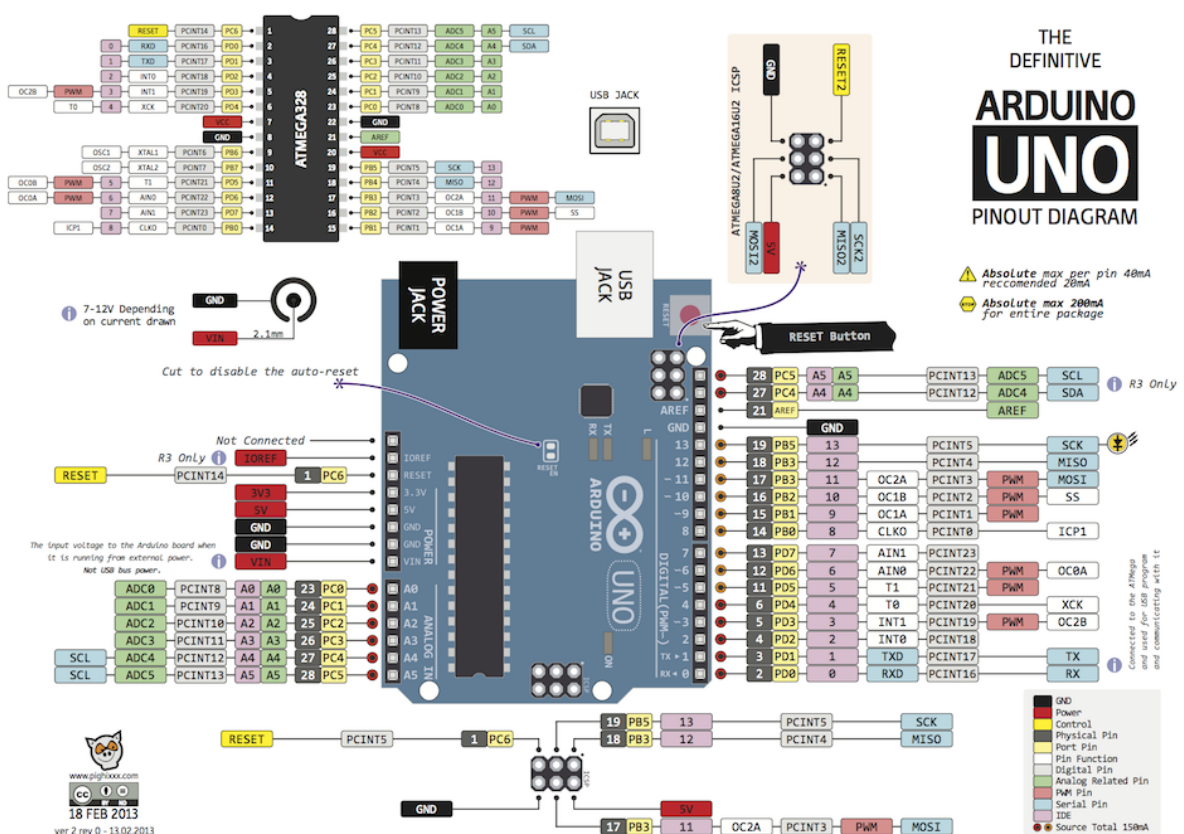


Figura 3.7 - Arquitetura do Arduino (Fonte: www.pighxxx.com).

### 3.5.2 – *Shield Ethernet*

O Arduino Ethernet ShieldW5100 ilustrado na Figura 3.8, baseia-se no chip WIZnet Ethernet W5100 que fornece comunicação à rede através da pilha de protocolos TCP/IP, suporta TCP, UDP, IPv4, ICMP, ARP, IGMP e PPPoE é utilizado usando a biblioteca Ethernet Library e SD Library. Para utilizar este Shield é preciso encaixá-lo na placa Arduino e plugá-lo na rede com um cabo de rede utilizando a entrada RJ45 através desse procedimento é possível controlar o Arduino através da Internet/Intranet.

É compatível tanto com o Arduino UNO e MEGA, possui um slot para cartão micro-SD que pode ser utilizado para armazenar arquivos.

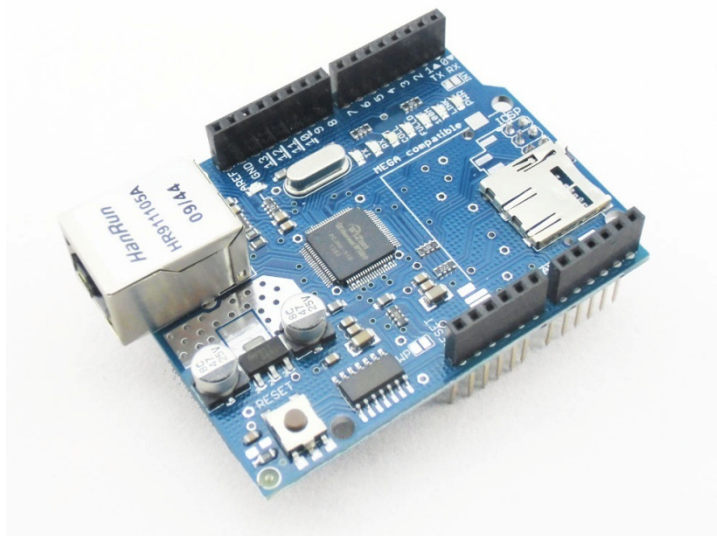


Figura 3.8 - *Shield Ethernet* WIZet W5100 (Fonte: [www.arduinoecia.com.br](http://www.arduinoecia.com.br)).

## 3.6 – Componentes eletrônicos e sensores utilizados no projeto

### 3.6.1 – Sensor de Temperatura e umidade DHT11

O sensor DHT11 (Figura 3.9) é um sensor de temperatura e umidade com uma sinal digital em sua saída calibrado. Sua tecnologia garante excelente estabilidade e confiabilidade. Um microcontrolador de alta performance de 8-bit garante isto no encapsulamento do módulo. Este sensor inclui um elemento resistivo do tipo NTC que faz a medição da temperatura. Possui excelente qualidade, resposta rápida, habilidade de anti-interferência e vantagens antes só encontradas em dispositivos de alto custo.

Cada sensor DHT11 apresenta características extremamente precisas de calibração em sua câmara de umidade. Os coeficientes de calibração são armazenados na memória do

programa OTP. O tamanho pequeno, o baixo consumo de energia e distância do sinal de transmissão de até 20 metros, torna-o interessante para uma variedade de aplicações e até mesmo em aplicações mais exigentes.

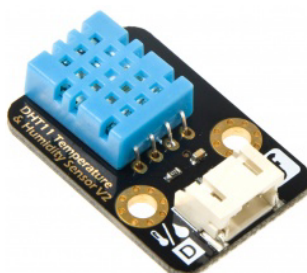


Figura 3.9 - Sensor DTH11 (Fonte [www.robocore.net](http://www.robocore.net)).

### 3.6.2 – Transistor TIP120

O transistor foi desenvolvido nos laboratórios da Bell em Murray Hill, New Jersey, Estados Unidos, em 1947 pelos cientistas: John Bardeen, Walter Brattain e William Shockley. O desenvolvimento desse componente semicondutor foi de grande relevância para a história da eletrônica e da informática, pois ele está presente em inúmeras invenções eletroeletrônicas, modificando vertiginosamente nossa sociedade. O transistor está na lista da IEEE de marcos na eletrônica, os inventores foram premiados com o prêmio Nobel de Física em 1956 por sua realização.

O termo “transistor” é a contração de duas palavras em inglês: *transfer resistor* (resistor de transferência). Existem dois tipos básicos de transistores de acordo com o tipo de dopagem de cada terminal (base, coletor e emissor), NPN e PNP. Um transistor é um dispositivo semicondutor usado para amplificar ou comutar sinais eletrônicos e energia elétrica. (Figura 3.10).

Um transistor é composto basicamente por três filamentos, chamados de base, emissor e coletor. O emissor é o polo positivo, o coletor, o polo negativo, enquanto a base é quem controla o estado do transistor que pode estar ligado ou desligado. Quando o transistor está desligado, não existe carga elétrica na base, por isso, não existe corrente elétrica entre o emissor e o coletor. Quando é aplicada certa tensão na base, o circuito é fechado e é estabelecida a corrente entre o emissor e o receptor.

Cada transistor funciona como uma espécie de interruptor, que pode estar ligado ou desligado, como uma torneira que pode estar aberta ou fechada, ou mesmo como uma válvula. A diferença é que o transistor não tem partes móveis como uma torneira e é muito menor, mais barato, mais durável e muito mais rápido que uma válvula.

A mudança de estado de um transistor é feita através de uma corrente elétrica. Cada



mudança de estado pode então comandar a mudança de estado de vários outros transistores ligados ao primeiro, permitindo o processamento de dados. Num transistor, essa mudança de estado pode ser feita bilhões de vezes por segundo, porém, a cada mudança de estado é consumida certa quantidade de eletricidade, que é transformada em calor.

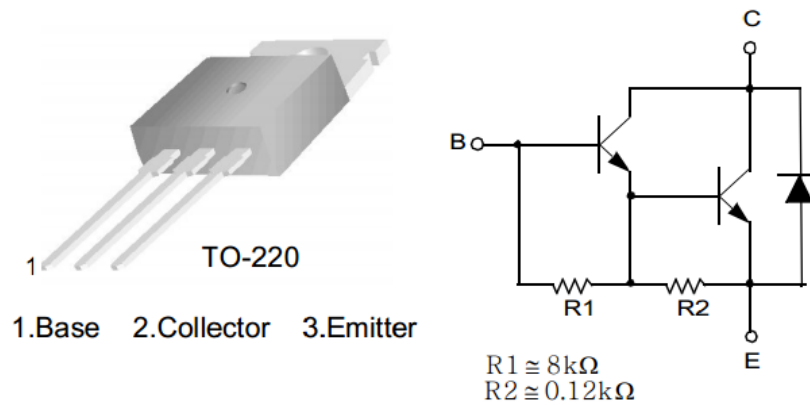


Figura 3.10 - Transistor do tipo NPN TIP120 (Fonte: *STMicroelectronics*).

### 3.6.3 – Sensor de Luminosidade LDR

A célula fotocondutora também é conhecida como LDR (Light Dependent Resistor, Resistor Dependente de Luz). É um tipo de semicondutor e a sua condutividade muda com variação proporcional à intensidade da luz. (Figura 3.11).

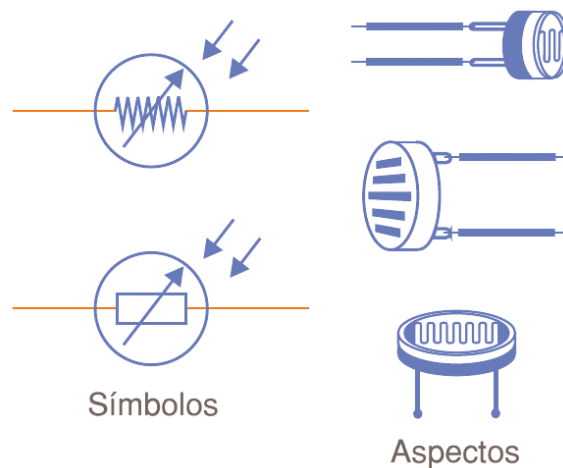


Figura 3.11 - LDR (Fonte: [www.sabereletronica.com.br](http://www.sabereletronica.com.br)).

Em um semicondutor existe uma diferença de energia entre os elétrons de condução e os elétrons de valência. Um LDR trata-se de um resistor dependente da luz, quando a luz



incide sobre ele, um fóton é absorvido, e consequentemente excita um elétron da banda de valência para a banda de condução. Devido a esses novos elétrons chegando à área da banda de condução, a resistência elétrica do dispositivo diminui. Assim, o LDR ou fotocondutor transdutor tem a resistência, que é o inverso da função de intensidade da radiação.

$$\lambda_0 = \frac{h.c}{e.E_\omega}$$

Devemos notar que toda a radiação com comprimento de onda maior do que o valor obtido na equação acima não pode produzir qualquer alteração na resistência deste dispositivo.

Vale ressaltar que um LDR tem uma desvantagem; quando sua temperatura é alterada, sua resistência muda drasticamente para uma intensidade de luz particular. Assim, este dispositivo não é adequado para medições precisas em aplicações analógicas.

#### 3.6.4 – Resistor

A resistência elétrica depende do material, das dimensões do condutor e da temperatura (agitação térmica). Sua unidade de medida no SI é o ohm. Essa unidade foi adotada em homenagem ao cientista alemão George Simon Ohm, que formulou a lei relacionando tensão, resistência e corrente elétrica em um elemento de circuito, representado pelo símbolo  $\Omega$ . Em muitos casos práticos, deseja-se que o valor da resistência seja o menor possível, para reduzir a dissipação de energia – por exemplo, nos condutores empregados em redes elétricas, transformadores e motores. Em outras aplicações, como nos circuitos eletrônicos, deseja-se limitar a corrente em um valor estipulado. Nesse caso, utiliza-se um componente especialmente destinado a esse fim, o resistor. Trata-se de um elemento físico cuja característica principal é a resistência elétrica.

Os resistores podem ser fixos ou variáveis. Neste caso são chamados de potenciômetros ou reostatos. O valor nominal é alterado ao girar um eixo ou deslizar uma alavanca. Os potenciômetros variam sua resistência máxima de acordo com cada modelo, porém o mais comum deles é o de  $10k\Omega$ . (Figura 3.12).



Figura 3.12 - Resistor e Potenciômetro (Fonte: [www.robocore.net](http://www.robocore.net)).

### 3.6.4 – Fans e Coolers

As *fans* ou exaustores são dispositivos que interagem com outros componentes do sistema de refrigeração elas atuam com ou sem um dissipador de calor, esse conjunto é comumente conhecido como *Cooler* (fan e dissipador) que seria o conjunto de uma ventoinha juntamente com o dissipador de calor representado na Figura 3.13. No projeto não será utilizado o dissipador, somente as *fans*.

As ventoinhas atuam expulsando o ar aquecido de determinado ambiente. Nos servidores existem dissipadores de calor e fileiras de *fans* (Figura 3.14) fabricados e desenhados de acordo com cada fabricante e são projetados para retirar o calor dos processadores, placas de expansão, memórias e HD's de alto desempenho.

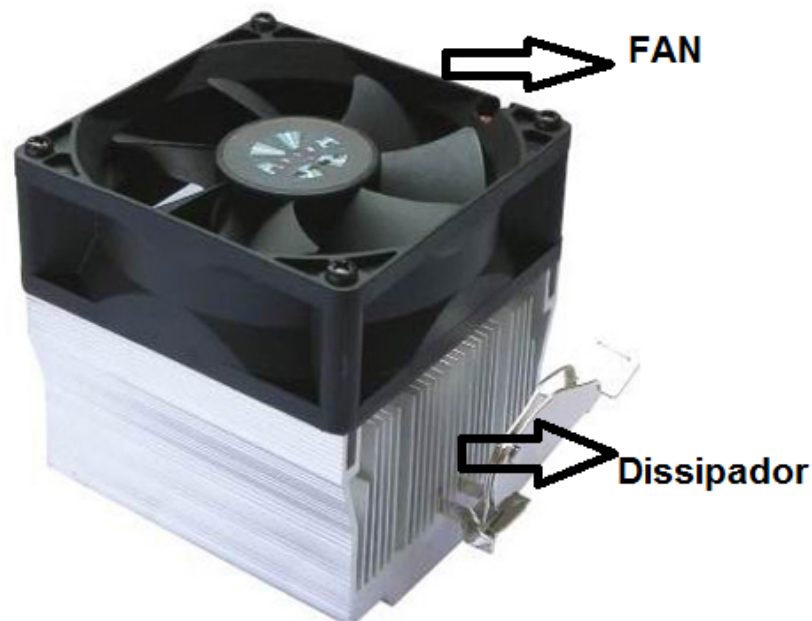


Figura 3.13 - Fan utilizado no projeto modelo: 60x60x25mm-Berflo com dissipador(Fonte: [spon.com.br](http://spon.com.br)).

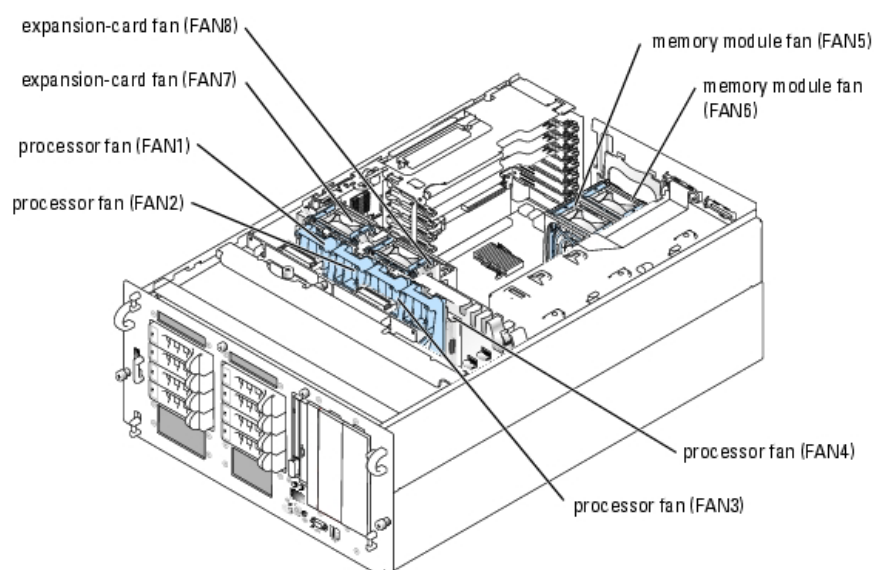


Figura 3.14 - Servidor DELL Power Edge (*Fonte: en.community.dell.com*).

### 3.7 – Linguagens de Programação Utilizadas no Projeto

#### 3.7.1 – Linguagem C

A linguagem do Arduino é derivada da linguagem C que é uma das linguagens de programação mais usadas de todos os tempos, e estão disponíveis para a maioria das arquiteturas de computadores e sistemas operacionais.

Desenvolvida por Dennis Ritchie durante o começo dos anos 70 para ser utilizada na implementação de sistemas operacionais e outras tarefas de programação de baixo nível. Ela foi derivada de outras duas linguagens anteriores: a Linguagem BCPL e a Linguagem B. A Linguagem BCPL foi desenvolvida por Martin Richards. Esta linguagem influenciou a linguagem inventada por Ken Thompson, chamada B. Logo em seguida, Dennis Ritchie desenvolveu a Linguagem C utilizando o sistema operacional UNIX. [KERNIGHAN e RITCHIE, 1998]

A linguagem de programação C começou sua marcha para se tornar formalmente definida pelo *American National Standard Institute* (ANSI), com a formação do comitê X3J11 em 1983. O trabalho da comissão foi concluído e o padrão aprovada em 1989. Desde então, a linguagem é muitas vezes referido como "ANSI C". O padrão também é reconhecido pela *International Organization for Standardization* (ISO), também, por isso às vezes ouve-se referência ao "ISO C". Para todos os efeitos, ANSI C e ISO C tem praticamente o mesmo significado. Também pode-se ouvir as duas versões serem chamadas de "*Standart C*".

No Arduino algumas funcionalidades do "*Standart C*" estão faltando o que não

impacta em nenhum problema, pois é utilizado um subconjunto do "Standard C" que já é mais do que capaz de realizar praticamente qualquer tarefa e as funcionalidades não encontradas podem ser contornadas de uma forma menos elegante.

### 3.8 – Meio de Transmissão das Informações

#### 3.8.1 – Protocolo TCP/IP

No final da década de 1960, a Agência ARPA (Advanced Research Projects Agency) iniciou o projeto de uma rede de computadores conhecida como ARPANET. Com o objetivo de implementar a rede física de suporte às comunicações e uma arquitetura de comunicação e seus respectivos protocolos.

Uma das primeiras arquiteturas de comunicação que surgiu foi uma arquitetura de 4 camadas desenvolvida pelo departamento de defesa (DOD - *Department of Defense*). O DOD serviu de base para a arquitetura mais utilizada atualmente, a arquitetura TCP/IP.

A arquitetura de comunicação TCP/IP inclui um conjunto de protocolos de suporte à transmissão de dados. A simplicidade e o fato de ser uma arquitetura aberta tornaram o TCP/IP na arquitetura mais utilizada na maioria das redes de comunicação de dados, incluindo a rede Internet. O TCP/IP utiliza conceitos e modelos similares aos do modelo OSI com estruturas de camadas bastante próximas. (Figura 3.15).

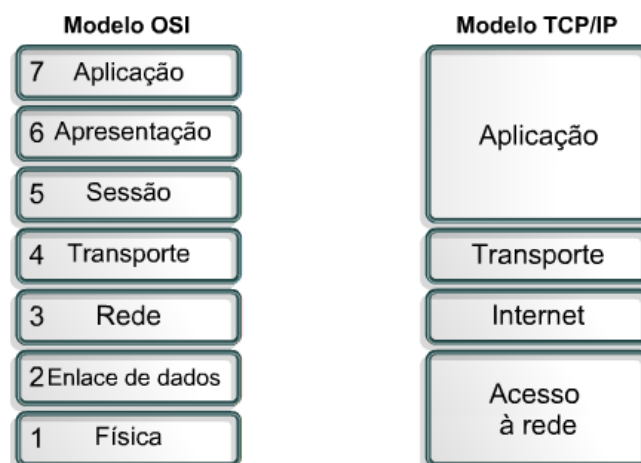


Figura 3.15 - Modelo de comunicação OSI e modelo TCP/IP (Fonte: Cisco Academy).

As camadas de aplicação do modelo TCP/IP são em geral comparadas às três camadas superiores do modelo OSI. Assim, os serviços de compressão, encriptação, administração de sessões e outros, são todos realizados na camada de aplicação.

A camada de transporte, de internet, de acesso à rede física têm serviços idênticos às camadas de transporte, rede, ligação à rede e física do modelo OSI, respectivamente.

A arquitetura TCP/IP é formada hoje por um vasto conjunto de protocolos que se distribuem pelas diferentes camadas do modelo.

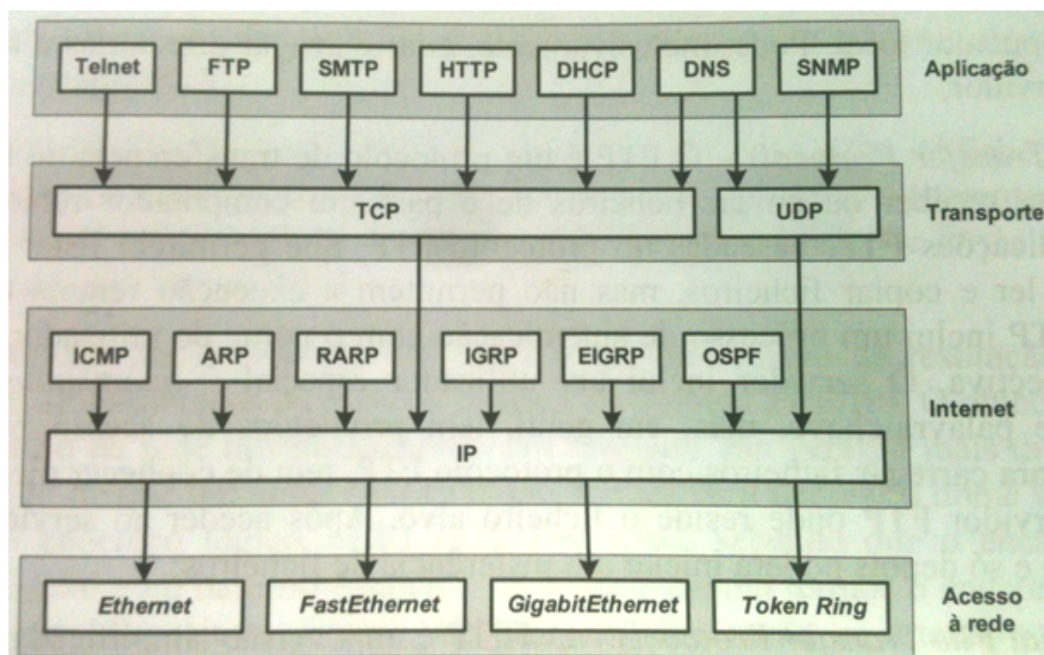


Figura 3.16 - Protocolo TCP/IP e suas camadas (*Fonte: Redes Cisco, Véstias 2009*).

A camada de aplicação inclui um vasto conjunto de aplicações ou protocolos de suporte a diversos serviços de comunicação, como a transferência de dados, a troca de e-mails, o acesso remoto, a troca de informações via *Web*, a gestão da rede, entre outros.

Um dos protocolos utilizados no projeto será o HTTP. O HTTP (Hypertext Transfer Protocol) é um dos protocolos mais utilizados para se ter acesso aos conteúdos das páginas *Web*. Quando o navegador *Web* envia uma requisição HTTP relativo à uma determinada página. Do outro lado o servidor *Web* responde ao pedido enviado a respectiva página.

## CAPÍTULO 4 – MODELO PROPOSTO

O Modelo proposto foi desenvolvido com o intuito de demonstrar que é possível eliminarmos um dos maiores problemas existente na maioria dos Datacenters que é o gasto expressivo com refrigeração e evitar os pontos de concentração de calor.

### 4.1 – Apresentações Gerais do Modelo Proposto

No fluxograma geral do sistema abaixo pode-se visualizar de uma forma abrangente todas as partes que fazem parte do processo de idealização e construção do protótipo.



Figura 4.1 - Fluxograma Geral do Modelo (*Fonte: Autor*).

### 4.2 – Descrições das Etapas do Modelo

O Esquemático da figura do projeto serviu de esboço para a criação e implementação do protótipo, com ele pode-se analisar todos os recursos necessários e onde seriam alocados os componentes dentro do receptáculo. Na Figura 4.2 observa-se os componentes já todos conectados. O Arduino é conectado à porta USB para fornecer à alimentação de 5V e o programa já funcionando fará com que as *Fans* alterem a sua velocidade de acordo com o dado relacionado à temperatura coletada pelo sensor de temperatura. As fontes de 12V alimentam as *Fans* e o transistor. O transistor irá aquecer à medida em que se diminui a resistência do potenciômetro, simulando o aquecimento de um servidor fazendo com que as *Fans* aumentem a sua velocidade de rotação para diminuir a temperatura. O LDR captará dados

O programa utilizado foi o Fritzing (<http://fritzing.org>) bastante útil para que se tenha o controle de todas as conexões.

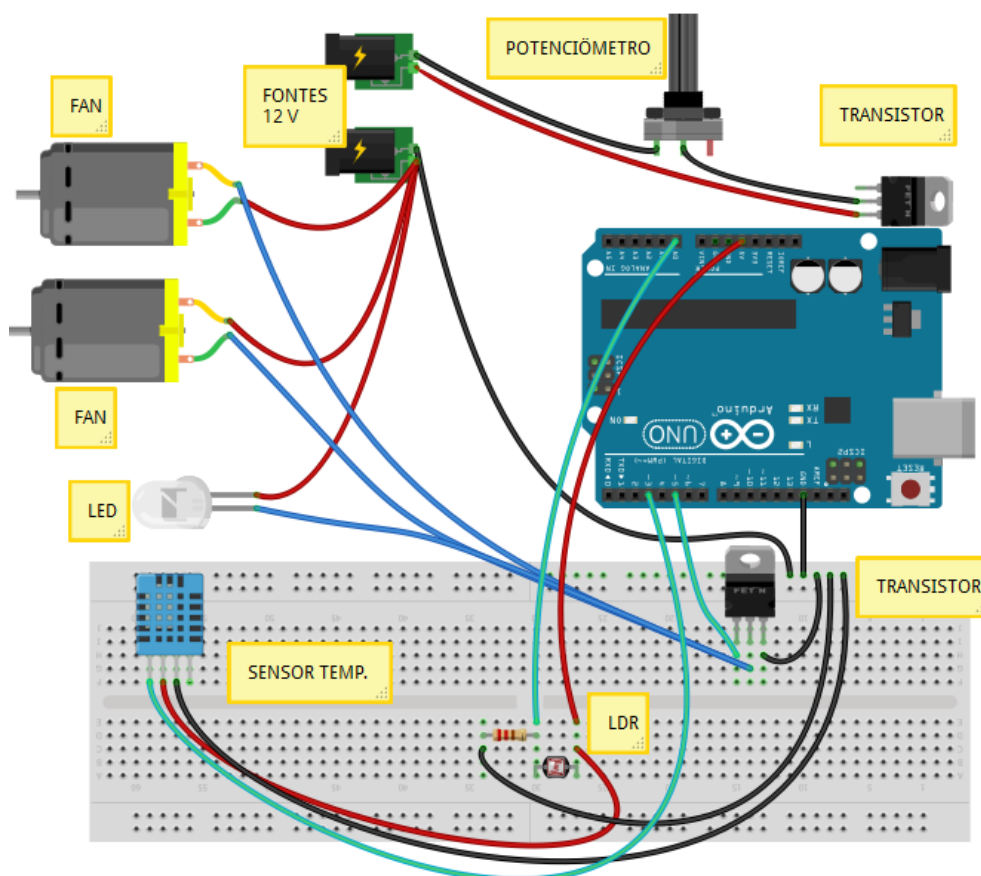


Figura 4.2 - Esquemático do Projeto (Fonte: Autor).

Com o esquemático do projeto já idealizado em sua fase final e visualizando cada parte que o compõe iniciou-se a elaboração do protótipo. Inicialmente definiu-se os pinos seriam utilizados por cada dispositivo de acordo com a figura 4.3. O LDR é o sensor de luminosidade, o DHT11 o sensor de temperatura e o TIP120 o transistor.

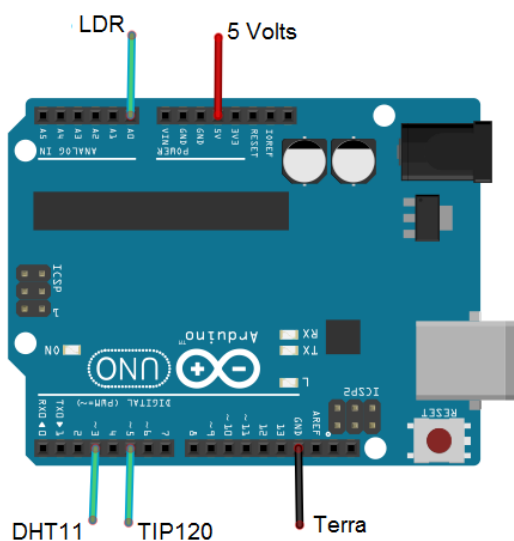


Figura 4.3 - Pinagem do Arduino utilizada (Fonte: Autor).



Em seguida acoplou-se a *Shield Ethernet* (ver item 3.5.2) no Arduino para que o microcontrolador pudesse interagir com a entrada RJ-45 e posteriormente com a Intranet.

A Figura 4.3 demonstra o Arduino já conectado à *Shield Ethernet*.

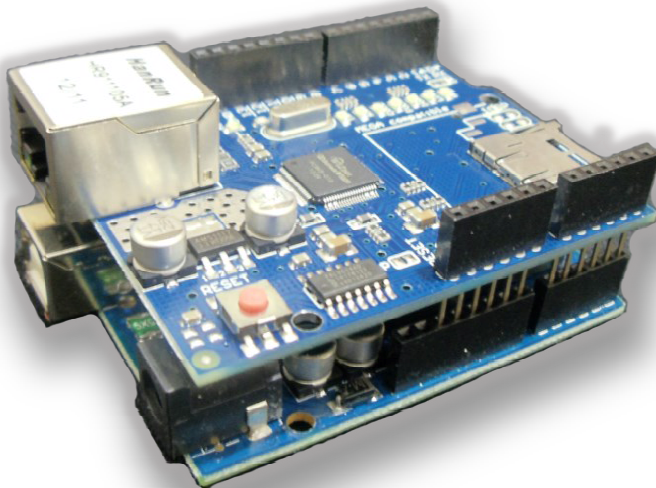


Figura 4.3- Representação da *Shield Ethernet* acoplado ao Arduino (*Fonte: Autor*).

Depois iniciou-se o processo de acomodação e encaixe das *fans* em uma pequena caixa juntamente com o protoboard, a *Shield Ethernet* já acoplada ao Arduino e as fontes de alimentação. Figura 4.4.

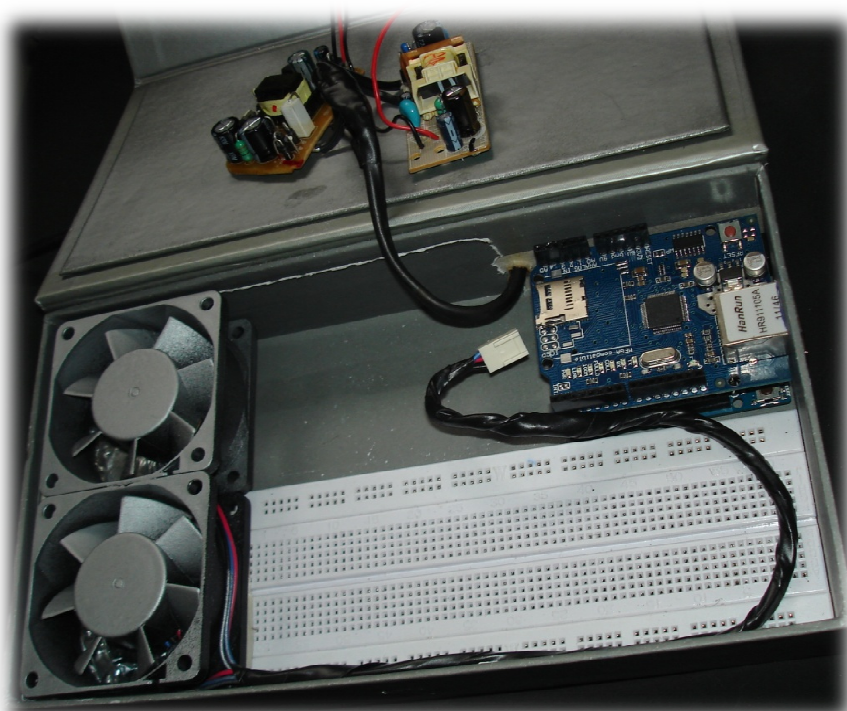


Figura 4.4 - Início da montagem do protótipo (*Fonte: Autor*).



Feito isso iniciou-se para a etapa de ligação dos fios respeitando as pinagens previamente definidas e depois os demais componentes do projeto também foram interligados conforme observado na Figura 4.5 que é o protótipo totalmente pronto.

A Pinagem no esquemático utilizando o Arduino segue a mesma lógica de pinagens na *Shield Ethernet*, ou seja, utiliza-se os mesmos pinos após o acoplamento do novo dispositivo. A *Shield Ethernet* obedece a mesma identificação do Arduino evitando uma possível troca de portas.

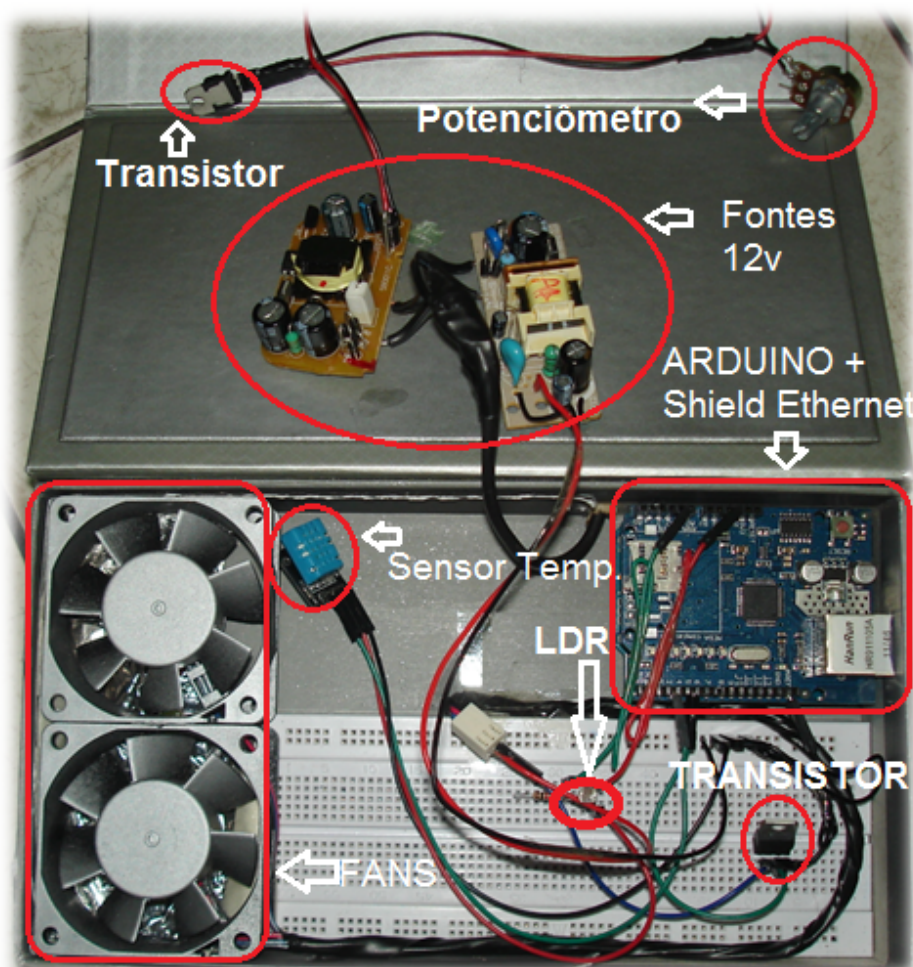


Figura 4.5 - Montagem final do protótipo (Fonte: Autor).

### 4.3 - Descrição da Implementação

De posse do circuito todo montado, efetuou-se a fase de compilação do código do projeto utilizando a interface de desenvolvimento (IDE) e posteriormente foi feito o *upload* desse código descrito no Apêndice para dentro do Microcontrolador.

O Projeto utiliza uma técnica bastante importante na eletrônica que é o PWM (*Pulse Width Modulation*, Modulação por largura de pulso). Através dessa técnica pode-se obter resultados analógicos com meios digitais. O controle digital é usado para criar uma onda quadrada. Essa onda é um sinal que se alterna entre ligado (5Volts) e desligado (0Volts)

Na figura 4.5 as linhas verdes representam um período de tempo regular. Esta duração ou período é o inverso da frequência de PWM. Em outras palavras, com a frequência PWM do Arduino em cerca de 500Hz, as linhas verdes medem 2 milissegundos cada. Uma chamada para `analogWrite ( )` é numa escala de 0 a 255, de tal modo que `analogWrite (255)` solicita um ciclo de trabalho de 100% (sempre) e `analogWrite (127)` é um ciclo de trabalho de 50% (em metade do tempo).

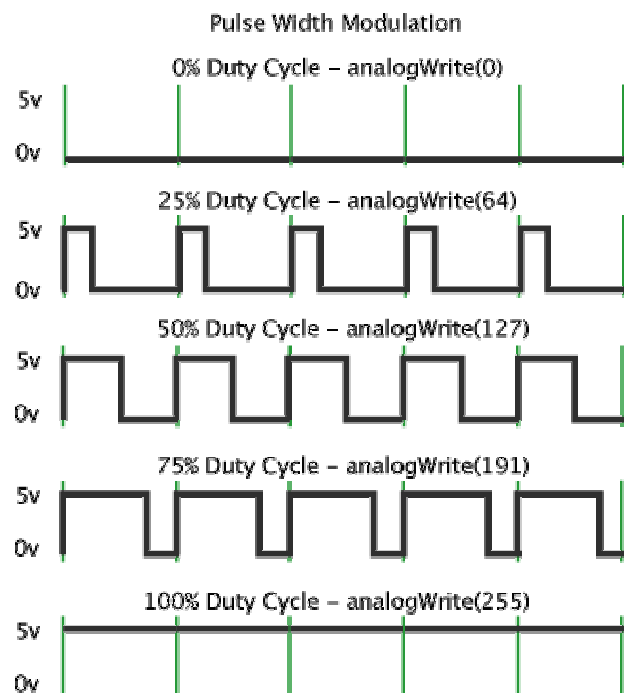


Figura 4.5 - PWM (Fonte: arduino.cc).

No código fonte utilizaremos o PWM da seguinte forma:

- `pwm = map(t, 15, 45, 15, 255);`
- `analogWrite(fan, pwm);`

Com isto, pode-se dizer que a variável “pwm” irá receber valores mapeados da seguinte forma: Variável Recebedora = `map(Valor Lido da Temperatura, Mínimo da Temperatura, Máximo da Temperatura, Mínimo da saída Analógica, Máximo da saída Analógica)`. Em suma o que irá ocorrer é a alteração da velocidade das *fans* proporcionalmente à temperatura lida pelo sensor.

## CAPÍTULO 5 - APLICAÇÃO PRÁTICA DO MODELO PROPOSTO

### 5.1 - Apresentação da área de Aplicação do modelo

A área de aplicação do modelo é projetada para ambientes de Datacenter onde necessita-se de um controle mais refinado da solução de resfriamento.

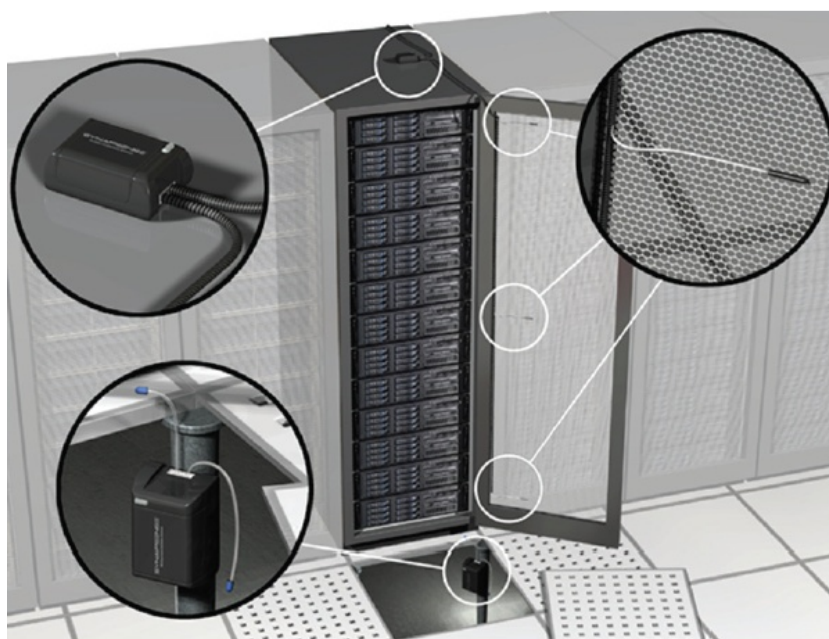


Figura 5.1- Wireless Sensor (Fonte:U.S. DepartamentOf Energy).

A ilustração Figura 5.2 demonstra de que forma pode ser implementado o modelo proposto. O modelo com vários sensores e *fans* permite uma regulagem de temperatura mais precisa.

### 5.2 – Descrição da Aplicação do Modelo

O modelo criado pode ser utilizado conforme ilustrado na Figura 5.2 onde representa-se a *computer room* dentro de um Datacenter e os servidores distribuídos nos *racks* evidenciam em vermelho os pontos de concentração de calor.

Os dados colhidos são enviados ao Servidor Web conforme figura 5.2; o tratamento dos valores relacionados à umidade e luminosidade também foram coletados pelos sensores apesar de não estarem dentro do escopo do projeto, servem para demonstrar que inúmeras possibilidades também podem ser exploradas como por exemplo a regulagem de um sistema de umidificação.

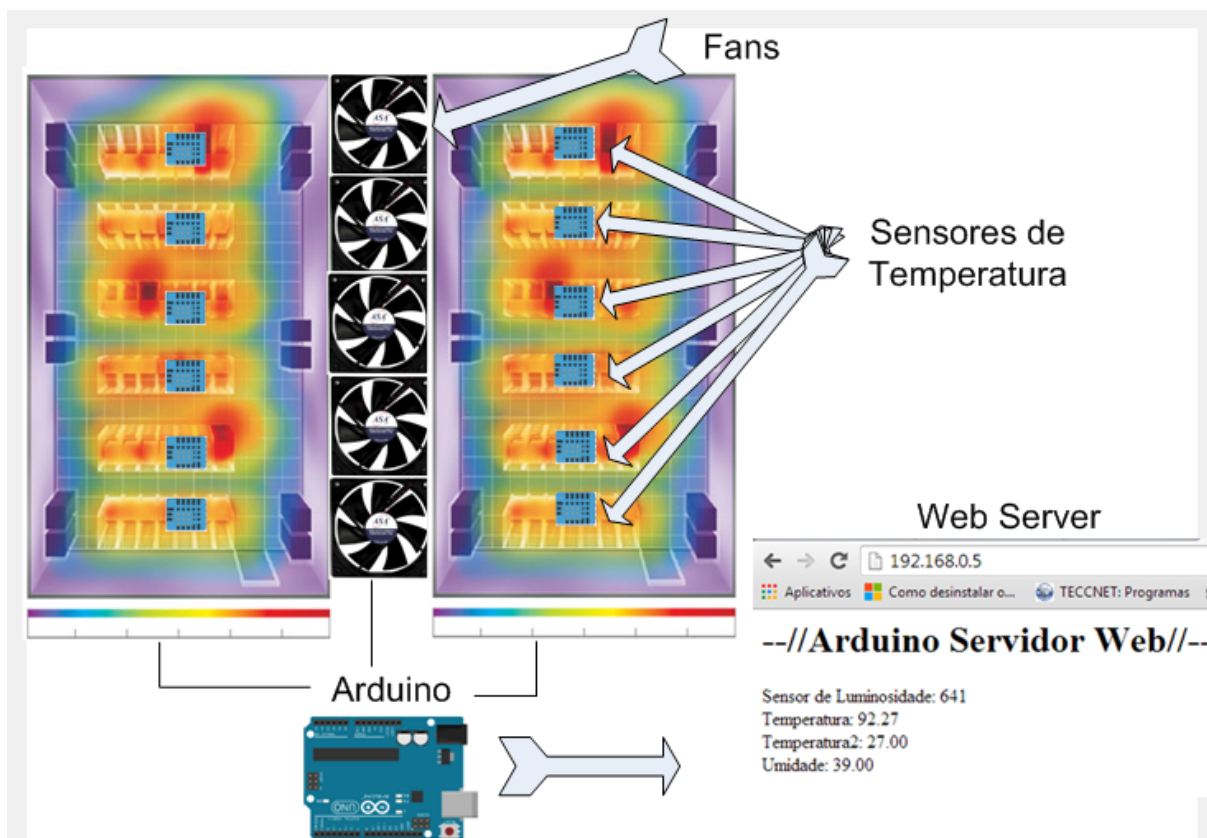


Figura 5.2 - Representação do protótipo de como poderia ser implementado em um Datacenter (Fonte:Autor).

### 5.3 – Resultados da Aplicação do Modelo

Após aquecimento do sensor observou-se que o Modelo demonstrou-se eficiente na velocidade de retirada do ar com auxílio das *Fans*. O sensor de temperatura demorou mais tempo para se adequar à temperatura ambiente sem o auxílio das *Fans*. Observou-se também que a temperatura ambiente interfere diretamente na velocidade de adequação da temperatura.

Sendo assim de posse de um dos levantamentos dos dados colhidos estimou-se a comparação conforme Tabela 5.1.

Tabela 5.1 – Velocidade da queda de temperatura.

TEMPERATURA	COM AUXÍLIO DAS VENTONHAS	SEM AUXÍLIO DAS VENTONHAS
Entre 40° e 45°	50 segundos.	5 minutos.

Fonte: Autor.

#### 5.4 – Custos do modelo proposto

Tabela 5.2 – Custo do Modelo.

<b>Componentes</b>	<b>Preços</b>
<i>Shield Ethernet</i>	R\$: 65,00 reais
Arduino	R\$: 70,00 reais
Fans (2x)	R\$: 15,00 reais
Componentes Eletrônicos(Led, Transistor, Resistor, Fios e Potenciômetro)	R\$: 20,00 reais
Sensor DTH11	R\$: 15,00 reais
Protoboard	R\$: 15,00 reais
<b>TOTAL</b>	<b>R\$: 200,00 reais</b>

Fonte: Autor.

#### 5.3 – Avaliação Global do Modelo

Como não foi possível a implementação em um Datacenter de fato pode-se inferir que para auxiliar a temperatura no micro ambiente com concentração de calor o protótipo precisa se adequar a estrutura encontrada em um determinado Datacenter dada a peculiaridade e complexidade de cada ambiente.

Um fator importante, é que todo o projeto foi elaborado utilizando componentes de baixo custo, sendo assim, um produto bastante viável economicamente.

Outro fator importante a se considerar é que o protótipo pode torna-se mais eficiente quando se agrega outros métodos de gerenciamento do fluxo de ar dentro desses ambientes aumentando o desempenho de todo o sistema de refrigeração podendo alcançar excelentes níveis de economia.

## CAPÍTULO 6 - CONCLUSÃO

### 6.1 – Conclusões

O Objetivo foi satisfatório, o protótipo atendeu as expectativas em um ambiente próximo ao encontrado em um Datacenter.

Os Dados foram aferidos com precisão e disponibilizados em um servidor *Web* conforme planejado. Figura 5.2.

A simulação de aquecimento do ambiente demonstrou que utilização de ventoinhas ou *fans* para a retirada do calor do micro ambiente se torna mais rápida e eficiente ao invés de esperar que o calor seja dissipado sem o auxílio de qualquer componente o que pode vir a comprometer a estabilidade dos equipamentos de TI envolvidos em questão.

Um ponto a ser considerado é que um maior controle do PWM pode ser explorado alterando a frequência padrão do ATmega328 permitindo assim um ajuste mais fino da velocidade das *fans*.

### 6.2 – Sugestões para Trabalhos Futuros

Interação com o sistema de controle de velocidade das *fans* e ativação e desativação dos mesmos remotamente.

Utilização de ventiladores potentes de frequência variável e medição do gasto(kw/h) dos ventiladores de acordo com a variação de temperatura do ambiente.

Realizar o cálculo do CFD de um sistema de refrigeração em Datacenters.

Utilização dos dados coletados relativos à umidade para criar um sistema de desumidificação.

## REFERÊNCIAS

1. MARIN, Paulo S. **Data Centers Desvendando cada passo: conceitos, projeto, infraestrutura física e energética** - 1ª ed, Rio de Janeiro, Brasport, 2010.
2. VERAS, Manuel. **Datacenter: componente central da infraestrutura de TI** - 1ª ed, São Paulo, Editora Érica, 2011.
3. VÉSTIAS, Mário. **Redes CISCO Para Profissionais** - 5ª ed, Lisboa, FCA - Editora de Informática, 2010.
4. MORAZ, Eduardo. **Curso Essencial de Hardware**. São Paulo, Digeraty Books, 2006.
5. McRoberts, Michael. **Beginning Arduino** - 2ª ed. New York, Apress, 2013.
- 6 ALEXANDRE, João. **ARDUINO Cartilha para programação em C** - 1ª ed [www.revista do arduino.com.br](http://www.revista.do.arduino.com.br) Janeiro de 2012
6. KENNETH, G. Brill **Cooling Capacity Factor (CCF) Reveals Stranded Capacity and Data Center Cost Savings** - , Upsite Technologies, Inc. White Paper, 2013.
- 7 DIETRICH, Jay **O Datacenter verde** - IBM Systems and Technology Group. © Copyright IBM Corporation 2007
- 8: MILLER, Rich. **The Productization of the Data Center** <<http://www.datacenterknowledge.com/>> Último acesso: 15 de novembro de 213
9. <<http://www.datacenterjournal.com/>> Último acesso: 20 de novembro de 213
10. TURNER, Pitt ; SEADER, Jonhn **Tier Classifications Define Site Infrastructure Performance** - Uptime Institute White Paper 2008

11. Neil Rasmussen and Brian Standley *Cooling Strategies for IT Wiring Closets and Small Rooms* White Paper 68 Schneider Electric – Data Center Science Center

12 By Liz Marshall and Paul Bemis, Applied Math Modeling Inc., Concord, NH *Using CFD for Data Center Design and Analysis* January, 2011

13 SOARES, Edileuza **Data centers brasileiros vendem nuvem de forma errada, diz pesquisa** <<http://computerworld.uol.com.br/tecnologia/2013/08/30/data-centers-brasileiros-vende-nuvem-de-forma-errada-diz-pesquisa/>> Último acesso: 27 de outubro de 2013

14 ANDRADE, Daniel *.Temperature Sensor + Arduino* <<http://www.danielandrade.net/2008/07/05/temperature-sensor-arduino/>> Julho 2008.



## APÊNDICE

```

#include "DHT.h"
#define DHTPIN 3
#define DHTTYPE DHT11
#include <SPI.h>
#include <Ethernet.h>

byte mac[] = {
  0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
IPAddress ip(192,168,0,5);

EthernetServer server(80);

DHT dht(DHTPIN, DHTTYPE);

int cooler = 5;
int pwm = 0;
int ValorT = 0;
float tempC;
int reading;
int tempPin = A1;
const int LDR = A0;
int ValorLDR = 0;

void setup()
{
  Serial.begin(9600);
  Ethernet.begin(mac, ip);
  server.begin();
  Serial.print("server is at ");
  Serial.println(Ethernet.localIP());
  Serial.println("Sensor DHT11 TESTE!");
  Serial.println("Inicializando...");
  analogReference(INTERNAL);
  dht.begin();
  //(cooler, OUTPUT);
}

void loop()
{
  EthernetClient client = server.available();
  if (client) {
    Serial.println("new client");
    // an http request ends with a blank line
    boolean currentLineIsBlank = true;
    while (client.connected()) {
      if (client.available()) {
        char c = client.read();
        Serial.write(c);

```

```

if (c == '\n' && currentLineIsBlank) {
    // send a standard http response header
    client.println("HTTP/1.1 200 OK");
    client.println("Content-Type: text/html");
    client.println("Connection: close");
    client.println("Refresh: 2");
    client.println();
    client.println("<body>\n");
    client.println("<body>\n");
    client.println("<h1>--//Arduino Servidor Web/--</h1>");
    client.println("<!DOCTYPE HTML>");
    client.println("<html>");
    // output the value of each analog input pin
    float h = dht.readHumidity();
    float t = dht.readTemperature();

    ValorLDR = analogRead(LDR);
    ValorT = analogRead(t);
    pwm = map(t, 15, 45, 15, 255);
    analogWrite(cooler, pwm);
    reading = analogRead(tempPin);
    tempC = reading / 9.31;
    if (t >=45)
    analogWrite (cooler, 255);
        client.print("Sensor de Luminosidade: ");
        client.print(ValorLDR);
        //client.println("<br />");
        //client.print("Temperatura: ");
        //client.print(tempC);
        client.println("<br />");
        client.print("Temperatura2: ");
        client.print(t);
        client.println("<br />");
        client.print("Umidade: ");
        client.print(h);
        client.println("<br />");

        client.println("</html>");
        break;
    }
    if (c == '\n') {
        // you're starting a new line
        currentLineIsBlank = true;
    }
    else if (c != '\r') {
        // you've gotten a character on the current line
        currentLineIsBlank = false;
    }
}
}
// give the web browser time to receive the data

```

```
    delay(1);  
    // close the connection:  
    client.stop();  
    Serial.println("client disconnected");  
  }  
}
```